Universidade de Aveiro
2013

Departamento de Eletrónica,
Telecomunicações e Informática

EVARISTE LOGOTA

CONTROL DESIGN FOR MULTICAST-AWARE CLASS-BASED NETWORKS

ESQUEMA DE CONTROLO PARA REDES MULTICAST BASEADAS EM CLASSES

Universidade de Aveiro
2013

Departamento de Eletrónica,
Telecomunicações e Informática

EVARISTE LOGOTA

CONTROL DESIGN FOR MULTICAST-AWARE CLASS-BASED NETWORKS

ESQUEMA DE CONTROLO PARA REDES MULTICAST BASEADAS COM CLASSES

Tese apresentada às Universidades de Minho, Aveiro e Porto para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Eletrotécnica, no âmbito do doutoramento conjunto MAP-Tele, realizado sob a orientação científica da Doutora Susana Isabel Barreto de Miranda Sargento, Professora auxiliar do Departamento de Eletrónica, Telecomunicações e Informática (DETI) da Universidade de Aveiro, e co-orientação do Doutor Augusto José Venâncio Neto, Professor adjunto do Instituto de Informática da Universidade Federal de Ceará, Brasil, e colaborador no Instituto de Telecomunicações de Aveiro.

Apoio financeiro da Fundação para a Ciência e a Tecnologia - FCT através da bolsa SFRH / BD / 33443 / 2008 e do FSE no âmbito do III Quadro Comunitário de Apoio.

To God, who makes it possible,
and to my parents, who gave me life

o júri

presidente

Prof. Doutor Artur da Rosa Pires
Professor Catedrático, Universidade de Aveiro

Prof. Doutor Mário Serafim dos Santos Nunes
Professor Catedrático, Instituto Superior Técnico, Universidade Técnica de Lisboa

Prof. Doutor Rui Luís Andrade Aguiar
Professor Associado com Agregação, Universidade de Aveiro

Prof. Doutor Paulo Manuel Martins Carvalho
Professor Associado, Escola de Engenharia, Universidade do Minho

Profª. Doutora Susana Isabel Barreto de Miranda Sargento
Professora Auxiliar, Universidade de Aveiro (**Orientadora**)

Prof. Doutor Augusto José Venâncio Neto
Professor Adjunto, Universidade Federal do Ceará (**Coorientador**)

acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Susana Sargento, for her patient guidance, constant support, useful critiques, and above all, her friendship. I am also grateful to my co-supervisor, Prof. Augusto Neto, for his friendship and valuable advices.

My gratitude is also due to the members of the MAP-Tele doctoral programme scientific committee, and to the staff of the Intitute of Telecommunications of Aveiro, for providing excellent research environment and facilities to achieve this work.

To my friend Carlos Campos, I would like to give my special thanks for his great assistance on the simulation platforms of this research work.

I would like to really thank my family and relatives who were always supporting and encouraging me, with dedication and prayers, to press forward with this PhD.

I am particularly grateful to Joana do Bem, Dr. Pedro Neves, and their families, whose friendship and hospitality in Portugal have been of great help for me over the years.

I would like to express my thankfulness and appreciation to Maura Outeiral, who was always there cheering me up and stood by me through the good and hard times of this work.

Special thanks are extended as well to Camila Quadros, Alfredo Matos, Georges Favraud, Abubakar Sadiq, Lucas Guardalben, and all my friends, colleagues and professors who assisted, advised, and supported my research and writing efforts. This Thesis would never have been completed without the devotion of my family and friends.

palavras-chave

Qualidade de Serviço, arquitetura e protocolos de rede, centralização, descentralização, sobre-aprovisionamento de recursos, recuperação a falhas, controlo de recursos e de admissão, reserva de recursos agregados.

resumo

À medida que as Tecnologias de Informação (TIs) se tornaram parte integrante da nossa sociedade, a expectativa dos cidadãos relativamente ao uso desses serviços também demonstrou um aumento, seja no âmbito das atividades profissionais, de lazer, aplicações de segurança crítica ou negócios. Portanto, as limitações dos projetos de rede tradicionais quanto ao fornecimento de serviços inovadores e aplicações avançadas motivaram um consenso quanto à integração de todos os serviços e infra-estruturas de comutação de pacotes, utilizando o IP, de modo a extrair benefícios económicos e um controlo mais flexível nas Redes de Nova Geração (RNG). Entretanto, tendo em vista que a Internet não apresenta capacidade de diferenciação de serviços, e sabendo que cada serviço apresenta as suas necessidades próprias, como por exemplo, a Qualidade de Serviço - QoS, a necessidade de formas mais evoluídas de comunicação tem-se tornado cada vez mais visível, levando a mudanças radicais na arquitectura das redes, que exigem soluções adequadas para a admissão de serviços e controlo de recursos de rede. Sendo assim, este trabalho aborda questões de controlo de QoS e rede com o objetivo de melhorar o desempenho do controlo de recursos total em redes atuais e futuras, através da análise dos serviços de acordo com as suas classes de serviço. Esta Tese encontra-se dividida em três partes.

Na primeira parte são propostos dois algoritmos de sobre-reserva, o Class-based bandwidth Over-Reservation (COR) e uma extensão melhorada do COR denominado de Enhanced COR (ECOR). A sobre-reserva significa a reserva de uma largura de banda maior para o serviço em questão do que uma classe de serviço (CoS) necessita e, portanto, a quantidade de sinalização para reserva de recursos é reduzida. COR e ECOR consideram uma definição dinâmica de sobre-reserva de parâmetros para CoSs com base nas condições da rede, com vista à redução da sobrecarga de sinalização em QoS sem que ocorra desperdício de largura de banda. O ECOR, por sua vez, difere do COR por permitir a otimização com minimização de controlo de overhead. Além disso, nesta Tese é proposto também um mecanismo de controlo centralizado chamado Advanced Centralization Architecture (ACA) , usando um único Ponto de Controlo de Decisão (CDP) que mantém uma visão ampla da topologia de rede e de análise dos recursos ocupados em tempo real como base de controlo para a rede global. Nesta Tese são utilizadas árvores multicast como base para o transporte de sessão, não só para fins de comunicação em grupo, mas principalmente para que os pacotes que pertençam a uma sessão que é mapeada numa determinada árvore sigam o seu caminho.

Os resultados obtidos nas simulações dos mecanismos mostram uma redução significativa da sobrecarga da sinalização de controlo, sem a violação dos requisitos de QoS ou desperdício de recursos. Além disso, foi proposto um modelo analítico no sentido de avaliar o impacto provocado por diversos parâmetros (como por exemplo, a capacidade da ligação, a dinâmica das sessões, etc), no sobre-provisionamento dos recursos.

Na segunda parte desta tese propõe-se um mecanismo para controlo descentralizado de recursos denominado de Advanced Class-based resource OverprRovisioning (ACOR), que permite obter uma melhor escalabilidade do que o obtido pelo ACA. O ACOR permite que os pontos de decisão e controlo da rede, os CDPs, sejam distribuídos na periferia da rede, cooperem entre si, através da troca de dados e controlo adequados (por exemplo, localização das árvores e informações sobre o uso da largura de banda), de tal forma que cada CDP seja capaz de manter um bom conhecimento da topologia da rede, bem como das suas ligações. Do ponto de vista de escalabilidade, a cooperação do ACOR é seletiva, o que significa que as informações de controlo são trocadas de forma dinâmica apenas entre os CDPs analisados. Além disso, a sincronização é feita através do conceito proposto de Recursos Virtuais Sobre-Provisionado (VOPR), que partilha as reservas de cada interface para cada árvore que usa a interface. Assim, cada CDP pode processar pedidos de sessão numa ou mais árvores, sem a necessidade de sincronização entre os CDPs correlacionados, enquanto o VOPR da árvore não estiver esgotado. Os resultados analíticos e de simulação demonstram que o controlo de sobre-reserva é agregado em cenários descentralizados, mantendo a sinalização de QoS baixa sem perda de largura de banda. Também é desenvolvido um protocolo de controlo de sinalização chamado ACOR Protocol (ACOR-P) para suportar as arquitecturas de centralização e descentralização deste trabalho. O ACOR Estendido (E-ACOR) agrega a VOPR de todas as árvores que se originam no mesmo CDP, e mais pedidos de sessão podem ser processados sem a necessidade de sincronização quando comparado com ACOR. Além disso, E-ACOR introduz um mecanismo para controlar as informações àcerca do congestionamento da rede, e impede a sincronização desnecessária durante o tempo de congestionamento quando os VOPRs esgotam consoante cada pedido de sessão. A avaliação de desempenho, através de resultados analíticos e de simulação, mostra a superioridade do E-ACOR em minimizar o controlo geral da carga da sinalização, mantendo todas as vantagens do ACOR, sem apresentar violações de QoS ou desperdício de recursos.

A última parte desta Tese inclui a proposta para recuperação a falhas, o Survivability ACOR (SACOR), o qual permite ter QoS estável em caso de falhas de ligações e nós. Os resultados de desempenho analisados mostram uma capacidade flexível de sobrevivência caracterizada por um tempo de convergência rápido e diferenciação de tráfego com uma utilização eficiente dos recursos.

Em resumo, os mecanismos de controlo de recursos propostos nesta Tese fornecem um suporte eficiente e escalável para controlo da rede, como também para os seus principais sub-sistemas (por exemplo, QoS, controlo de recursos, engenharia de tráfego, multicast, etc) e, assim, permitir a otimização do desempenho da rede a nível do controlo global.

keywords

Quality of Service, network architecture and protocols, centralization, decentralization, resource overprovisioning, survivability, resource and admission control, aggregate resource reservation.

abstract

The expectations of citizens from the Information Technologies (ITs) are increasing as the ITs have become integral part of our society, serving all kinds of activities whether professional, leisure, safety-critical applications or business. Hence, the limitations of the traditional network designs to provide innovative and enhanced services and applications motivated a consensus to integrate all services over packet switching infrastructures, using the Internet Protocol, so as to leverage flexible control and economical benefits in the Next Generation Networks (NGNs). However, the Internet is not capable of treating services differently while each service has its own requirements (e.g., Quality of Service - QoS). Therefore, the need for more evolved forms of communications has driven to radical changes of architectural and layering designs which demand appropriate solutions for service admission and network resources control. This Thesis addresses QoS and network control issues, aiming to improve overall control performance in current and future networks which classify services into classes. The Thesis is divided into three parts.

In the first part, we propose two resource over-reservation algorithms, a Class-based bandwidth Over-Reservation (COR) and an Enhanced COR (ECOR). The over-reservation means reserving more bandwidth than a Class of Service (CoS) needs, so the QoS reservation signalling rate is reduced. COR and ECOR allow for dynamically defining over-reservation parameters for CoSs based on network interfaces resource conditions; they aim to reduce QoS signalling and related overhead without incurring CoS starvation or waste of bandwidth. ECOR differs from COR by allowing for optimizing control overhead minimization. Further, we propose a centralized control mechanism called Advanced Centralization Architecture (ACA), that uses a single state-full Control Decision Point (CDP) which maintains a good view of its underlying network topology and the related links resource statistics on real-time basis to control the overall network. It is very important to mention that, in this Thesis, we use multicast trees as the basis for session transport, not only for group communication purposes, but mainly to pin packets of a session mapped to a tree to follow the desired tree. Our simulation results prove a drastic reduction of QoS control signalling and the related overhead without QoS violation or waste of resources. Besides, we provide a generic-purpose analytical model to assess the impact of various parameters (e.g., link capacity, session dynamics, etc.) that generally challenge resource overprovisioning control.

In the second part of this Thesis, we propose a decentralization control mechanism called Advanced Class-based resource OverpRovisioning (ACOR), that aims to achieve better scalability than the ACA approach. ACOR enables multiple CDPs, distributed at network edge, to cooperate and exchange appropriate control data (e.g., trees and bandwidth usage information) such that each CDP is able to maintain a good knowledge of the network topology and the related links resource statistics on real-time basis. From scalability perspective, ACOR cooperation is selective, meaning that control information is exchanged dynamically among only the CDPs which are concerned (correlated).

Moreover, the synchronization is carried out through our proposed concept of Virtual Over-Provisioned Resource (VOPR), which is a share of over-reservations of each interface to each tree that uses the interface. Thus, each CDP can process several session requests over a tree without requiring synchronization between the correlated CDPs as long as the VOPR of the tree is not exhausted. Analytical and simulation results demonstrate that aggregate over-reservation control in decentralized scenarios keep low signalling without QoS violations or waste of resources. We also introduced a control signalling protocol called ACOR Protocol (ACOR-P) to support the centralization and decentralization designs in this Thesis. Further, we propose an Extended ACOR (E-ACOR) which aggregates the VOPR of all trees that originate at the same CDP, and more session requests can be processed without synchronization when compared with ACOR. In addition, E-ACOR introduces a mechanism to efficiently track network congestion information to prevent unnecessary synchronization during congestion time when VOPRs would exhaust upon every session request. The performance evaluation through analytical and simulation results proves the superiority of E-ACOR in minimizing overall control signalling overhead while keeping all advantages of ACOR, that is, without incurring QoS violations or waste of resources.

The last part of this Thesis includes the Survivable ACOR (SACOR) proposal to support stable operations of the QoS and network control mechanisms in case of failures and recoveries (e.g., of links and nodes). The performance results show flexible survivability characterized by fast convergence time and differentiation of traffic re-routing under efficient resource utilization i.e. without wasting bandwidth.

In summary, the QoS and architectural control mechanisms proposed in this Thesis provide efficient and scalable support for network control key sub-systems (e.g., QoS and resource control, traffic engineering, multicasting, etc.), and thus allow for optimizing network overall control performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND SYMBOLS

A

3G/4G/5G	3rd Generation/4th Generation/5th Generation
3GPP	Third Generation Partnership Program
A-RACF	Access-Resource and Admission Control Function
AAA	Authentication, Authorization and Accounting
AC	Admission Control
ACA	Advanced Centralization Architecture
ACA-B	ACA-Border agent
ACA-F	ACA-Full agent
ACA-L	ACA-Light agent
ACM	Association for Computing Machinery
ACOR	Advanced Class-based bandwidth OverpRovisioning
ACOR-P	ACOR Protocol
AF	Assured Forwarding (class of service); Application Function (RACS)
AMAC	Active Measurement-based Admission Control
AN	Access Network
ARR	Available Reservation-based Re-routing
AS	Autonomous System
ASF	Application Support Functions
ATM	Asynchronous Transfer Mode
ATRF	Automatic Traffic Re-routing Functions

B

BE	Best Effort
BF	Basic Functions
BGF	Border Gateway Function
BGMP	Border Gateway Multicast Protocol
BGP	Border Gateway Protocol
BGRP	Border Gateway Reservation Protocol
BR	Border Router
BTF	Basic Transport Function

C

C-CAST	Context Casting
CASM	Cache-based Seamless Mobility
CBR	Constant Bit Rate
CBT	Core Based Tree
CCAMP	Common Control and Measurement Plane
CDF	Content Delivery Function
CDP	Control Decision Point
CL	Controlled Load service
COPS	Common Open Policy Service
COPS-PR	COPS for support of policy provisioning

COR	Class-based bandwidth OverpRovisioning
CoS	Class of Service
CPU	Central Processing Unit
CR-LDP	Constraint-based Routing Label Distribution Protocol
CS	Control Signalling CoS
CSCF	Call Session Control Function
CXT_SPEC	ConteXT information SPECification

D

DAIDALOS	Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services
DARIS	Dynamic Aggregation of Reservations for Internet Services
DESF	Down Events Synchronization Functions
DiffServ	Differentiated Services
DR	Designated Router
DS	Differentiated Service domain
DSCP	Differentiated Services Code Point
DVB	Digital Video Broadcasting
DVMRP	Distance Vector Multicast Routing Protocol
DWDM	Dense Wave Division Multiplexing

E

ECF	Elementary Control Function
E-ACOR	Extended ACOR
ECOR	Enhanced Class-based bandwidth OverpRovisioning
EF	Expedited Forwarding
EFF	Elementary Forwarding Functions
ER	Edge Router
ETRF	Extra Traffic Re-routing Functions
ETSI	European Telecommunications Standards Institute
EUI-48	48 bits for IEEE Extended Unique Identifier
EUI-64	64 bits for IEEE Extended Unique Identifier

F

FE	Functional Entity
FEC	Forwarding Equivalence Class
FIB	Forwarding Information Base
FIFO	First In First Out
FTP	File Transfer Protocol

G

GIST	General Internet Signalling Transport
GMPLS	Generalized Multi-Protocol Label Switching
GS	Guaranteed Service

H

HSS	Home Subscriber Server

I

I-CSCF	Interrogating Call Session Control Function
IBCF	Interconnection Border Control Function
ID	Identification
IdM	Identity Management
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ILP	Integer Linear Programming
IMS	IP (Internet Protocol) Multimedia Sub-system
INFO_SPEC	Information Specification
IntServ	Integrated Services
IP	Internet Protocol
IPTV	IP Television
IPv4	IP Protocol version 4
IPv6	IP Protocol version 6
IS-IS	Intermediate System to Intermediate System
IS-IS-TE	Intermediate System to Intermediate System-Traffic Engineering
ISDN	Integrated Services Digital Network
ISM	Internet Standard Multicast
ISP	Internet Service Provider
ITU-T	International Telecommunication Union - Telecommunication

L

LDP	Label Distribution Protocol
LER	Label Edge Routers
LSA	Links States Advertisement
LSD	Label Switching Domain
LSP	Label Switched Path
LSR	Label Switch Router

M

MAC	Medium Access Control
MARA	Multi-user Aggregated Resource Allocation
MASC	Multicast Address Set Claim
MBGP	Multiprotocol Border Gateway Protocol
MBMS	Multimedia Broadcast Multicast Services
MF	Management Function
MIB	Management Information Base
MIRA	Multi-service Resource Allocation
MMCF	Mobility Management and Control Function
MOSPF	Multicast Open Shortest Path First
MPLS	MultiProtocol Label Switching
MPLS-TP	MultiProtocol Label Switching – Transport Profile
MRI	Message Routing Information

MRIB	Multicast Routing Information Base
MSDP	Multicast Source Discovery Protocol
MSPEC	Multicast Specification
MUSC	Multi-user Session Control

N

NACF	Network Attachment Control Functions
NAPT	Network Address and Port Translation
NASS	Network Attachment Sub-system
NAT	Network Address Translation
NetCIB	Network Context Information Base
NGN	Next Generation Networks
ns-2	Network Simulation version 2
NSIS	Next Steps in Signalling
NSLP	NSIS Signalling Layer Protocol
NSLPID	NSLP Identifier
NTLP	NSIS Transport Layer Protocol

O

OSMAR	Overlay for Source-Specific Multicast in Asymmetric Routing
OSPF	Open Shortest Path First
OSPF-TE	Open Shortest Path First-Traffic Engineering

P

P-CSCF	Proxy Call Session Control Function
P2MP	Point-to-Multi-Point
PAC	Parameter-based Admission Control
PD-FE	Policy Decision Functional Entity
PDP	Policy Decision Point
PDR	Per Domain Reservation
PE-FE	Policy Enforcement Functional Entity
PEP	Policy Enforcement Point
PHB	Per-Hop Behaviour
PHR	Per Hop Reservation
PIM	Protocol Independent Multicast
PIM-DM	PIM-Dense Mode
PIM-SM	PIM-Sparse Mode
PIM-SSM	Protocol Independent Multicast for Source Specific Multicast
PMAC	Passive Measurement-based Admission Control
PR	Preemptive-based Re-routing
pSLA	provider SLA
pSLS	provider SLS
PWE3	Pseudo Wire Emulation Edge-to-Edge architecture

Q

Q3M	Multi-user Mobile Multimedia

QoS	Quality of Service
QoSM	QoS Model
QoS-NSLP	QoS Signalling Layer Protocol
QSPEC	QoS specification object

R

RAC	Resource and Admission Control
RACF	Resource and Admission Control Function
RACS	Resource and Admission Control Sub-system
RAO	IP Router Alert Option
RED	Random Early Detection
RC	Resource Control
RCEF	Resource Control Enforcement Function
RIB	Routing Information Base
RII	Response Identification Information
RIP	Routing Information Protocol
RMD	Resource Management in DiffServ
RMF	Resource Management Function
RODA	Resource Management in DiffServ On-demAnd
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RRO	Record Route Object
RRR	Reservation Readjustment-based Re-routing
RSpec	Reservation Specification
RSVP	Resource ReSerVation Protocol
RSVP-TE	Resource ReSerVation Protocol-Traffic Engineering
RTP	Real-Time Transport Protocol

S

S-CSCF	Serving Call Session Control Functions
SACOR	Survivable ACOR
SC	Synchronization Control
SCF	Service Control Function
SDH	Synchronous Digital Hierarchy
SDP	Session Description Protocol
SGM	Small Group Multicast
SICAP	Shared-segment Inter-domain Control Aggregation Protocol
SID	Session Identifier
SIDSP	Simple Inter-Domain QoS Signalling Protocol
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SLS	Service Level Specification
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol

SONET	Synchronous Optical Networking
SPDF	Service Policy Definition Function
SRBQ	Scalable Reservation-based QoS
SSF	Service Support Functions
SSM	Source Specific Multicast

T

TCA	Traffic Conditioning Agreement
TCF	Transport Control Functions
TCP	Transmission Control Protocol
TISPAN	Telecommunications and Internet Converged Services and Protocols for Advanced Networking
TLV	Type Length Value
TRC-FE	Transport Resource Control Functional Entity
TRE-FE	Transport Resource Enforcement Functional Entity
TSpec	Traffic Specifications

U

UDP	User Datagram Protocol
UE	User Equipment
UESF	Up Events Synchronization Functions
UMTS	Universal Mobile Telecommunications System

V

VC	Virtual Circuit
VoIP	Voice over IP
VOPR	Virtual Over-Provisioned Resource
VP	Virtual Path
VR	VOPR-based Re-routing

W

WDM	Wavelength-Division Multiplexing
WFQ	Weighted Fair Queuing
WiFi	Wireless Local Area Network - WLAN
WiMAX	Worldwide Interoperability for Microwave Access
WWW	World Wide Web

Chapter 1

Introduction

This Thesis deals with scalable QoS and network control for current and future class-based networks. We exploit inherent scalability feature of class-based control and propose new control mechanisms which allow for optimizing network overall performance by providing scalable resources control through descentralized network control with resources overprovisioning. The proposed system can be used and extended to support operations for QoS in multicast services, traffic engineering, routing, link capacity planning, network virtualization control and mobility.

This chapter is structured as follows. Section 1.1 presents our motivations for this research work. Section 1.2 describes our objectives and our main contributions to advance the state of the art in the area. Further, section 1.3 lists our publications and section 1.4 presents the organization of the Thesis.

1.1 *Motivations*

In contrast with what one could observe several years ago, there is nowadays a high expectation of innovative and attractive services in all human-centric aspects whether professional, leisure, health, safety, security and business. The real-time QoS stringent multimedia applications (e.g., Internet Protocol Television - IPTV, Videoconferecing, Online Games, etc.), personalized and immersive services (e.g., Facebook, Virtual meeting, etc.) are examples of these services. The citizens are willing to be able to select from a wide range of user experienced QoS, to get the information content they want, anywhere, anytime and over any facilities available. Moreover, service consumers are getting into the role of content creators (e.g., YouTube), while the advent of multihomed terminals (terminal with multiple interfaces) increased the need for QoS and mobility support across heterogeneous access technologies (e.g., Universal Mobile Telecommunications

System/Multimedia Broadcast Multicast Services - UMTS/MBMS, Worldwide Interoperability for Microwave Access - WiMAX, Wireless Local Area Network - WiFi, and Digital Video Broadcasting - DVB).

In this context, the NGN, as defined by the International Telecommunication Union – Telecommunication (ITU-T) [1], is *a packet-based network able to provide telecommunication services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent from underlying transport related technologies. It enables unfettered access for users to networks and to competing service providers and/or services of their choice. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users.*

However, besides flexibility and scalability features, the widely adopted packet-based Internet technology does not provide means for traffic differentiation control while each service has its own requirements (e.g., QoS). In other words, the legacy Internet only treats all services equally and thus raises major concerns for reconsideration of our traditional network design philosophy.

In order to provide QoS to each individual traffic flow in the Internet, the Internet Engineering Task Force (IETF) proposed the Integrated Services (IntServ) QoS architecture [2] which bases on fine-grained control principles such that, the network nodes on communication path are signalled upon service request to perform admission control and to reserve resources (e.g., bandwidth) for the session establishment. This implies that a service request may be accepted or rejected depending on the QoS requirements of the service and the available resource conditions in the network. Consequently, this approach confronts scalability and long session setup time issues since signalling is generated on per-flow basis and every router on the path must maintain reservation state, carry out scheduling functions and admission control for each flow.

Therefore, IETF introduced the Differentiated Services (DiffServ) QoS architecture as a scalable approach [3]. In DiffServ, traffic flows are classified into a limited number of CoSs at network entrance points (border/edge) and are treated aggregately without resource reservation signalling or admission control mechanism along a path. At each interior router in a DiffServ domain, packets are simply forwarded based on the CoS parameter (class Identification - class ID) available in the Internet Protocol (IP) packet header and the pre-configured forwarding treatment of the CoS. However, the Internet routing protocols mainly rely on shortest path first [4] to route traffic flows, which raises inefficient resource utilization problems, since some segments of network may be congested or even over-utilized while other segments may be under-utilized. Hence, the lack of appropriate control on paths exposes DiffServ to QoS degradation, where the

amount of resource required by the flows in a CoS may happen to exceed the affordable capacity of the CoS.

The approaches, such as IP over Asynchronous Transfer Mode (ATM) and IP over frame relay [5], cope with QoS as overlay control models. Basically, the overlays extend the traditional design space by enabling the provision of arbitrary virtual topologies on top of real network infrastructures (underlays). They provide explicit virtual paths, constraint-based routing, and robustness as additional functions to improve traffic and resource control. Thus, the IETF standardized the Multiprotocol Label Switching (MPLS) [6] which potentially provides most of the functionality available from the overlay models in an integrated and more cost-effective manner when compared to existing alternatives. Hence, the MPLS-based networks (e.g., Generalized MPLS - GMPLS [7], MPLS Transport Profile - MPLS-TP [8], [9]) offer the possibility to automate key control features such as traffic engineering aspects, which are indispensable functions to efficiently minimize unnecessary congestion occurrence, and to allow for improving resource and control performance over the Internet.

As such, network resource and admission control functions and protocols have been considered and intensively discussed as one of the key functionalities to improve networking performance [10]. It is broadly studied that a good knowledge of network topological information along with available resources [11] on bottleneck interfaces (interfaces with smallest available resources) and their location on paths [12], are of paramount importance to allow for improving control performance. In other words, a good view of underlying networks infrastructures (e.g., network nodes, interfaces capacities, etc.) is essential to improve performance [13]. The ITU-T G.1081 [14] defines five monitoring points in networks, allowing service providers to monitor networks and services performance in terms of resources utilization and optimization. In the literature, existing network monitoring or measurement proposals mostly employ paths' probing techniques [15], [16] to acquire knowledge of network resource utilization conditions. However, research results [17], [18], [19] show that paths' probing raises major performance concerns mainly in terms of heavy signalling overhead (depending on probing frequency), complexity, as well as accuracy issues. An example is that excessive signalling events can easily overwhelm socket Input/Output interfaces [20], and therefore, networking control design must be carefully addressed [21] in the NGN.

As a result, it is deemed crucial that QoS and resource reservation control be performed aggregately [22], meaning that a single signalling set can be used for a bulk or surplus of resource reservation for CoSs across a domain, so as to reduce the signalling rate and the related control state granularity. This is also known as dynamic aggregate resource over-reservation. However, network interfaces are usually shared by multiple paths originated at different sources, which

results in dynamic consumptions of network resources, especially in distributed control scenarios [23]. Hence, aggregate bandwidth reservation or over-reservation strongly requires a good knowledge of available bandwidth information in each CoS on the outgoing interfaces along paths on real-time-basis without signalling the paths as long as possible. In this sense, while the aggregation of reservations allows for reducing QoS control signalling and state overhead, the impact of resource fragmentation [24], [25] and waste of resources is a major trade-off since surplus of reservations may not be efficiently used, and service blocking probability would increase unnecessarily. Moreover, inefficient admission control, seeking optimization of resource utilization, can still place excessive signalling, especially under severe network conditions such as congestion periods of time.

Furthermore, IP Multicast [26] has been broadly investigated as the best promising technology for group communications such as multimedia sharing, collaboration between people, social networking over the Internet. Many proposals [27], [28], [29], [30] have sprung in the research community to provide QoS for multicast sessions. The designs started to incorporate logical and intelligent entities (e.g., tree manager, Multicast Controller, QoS Broker, etc.) that focus on QoS routing, admission control, resource reservation, group-to-tree matching, and policy control [31], [32], [33]. In [34], asymmetric routing problem is addressed by populating/updating Multicast Routing Information Base (MRIB) with the more suitable routes information, which is used by Protocol Independent Multicast for Source Specific Multicast (PIM-SSM) [35] to build the best distribution trees that data are to follow. However, while multiple multicast flows may be also aggregated [36] into a single flow through encapsulation mechanism [37], [38] to reduce multicast forwarding states information, the QoS control signalling is mostly per-flow based, and particular attention is still necessary for the sake of scalability.

The advances in fibre optic transport technology, offering high capacity, are expected to alleviate the effects of congestion. However, service demands and the requirements are also growing rapidly due to increasing dependence of citizens and businesses on the Information Technologies. In addition, the service demands and traffic behaviors are mostly unpredictable. As it is studied in [23], service admission control in a network cannot be planned; it should dynamically take into account current traffic demands and available network resources to achieve cost-effective performance. Therefore, the need for appropriate solutions and mechanisms for the networking and resource management prevailed, which drove to radical changes in the architectural layering control in the scope of the NGNs. In particular, it is expected that service-related functions must be completely decoupled from the underlying transport-related technologies. This way, applications can be developed independently from the underlying connectivity considerations and the transport control intelligence layer would provide support for all types of services and information over the

packet-based network. Basically, the transport will address traffic-oriented objectives in terms of QoS requirements (e.g., bandwidth, delay, jitter and packet loss), and resource-oriented objectives to allow for efficient management of the network resources. This implies that transport intelligence will be aware of the services (e.g., Voice over IP - VoIP, data, etc.) that they carry and be able to assure differentiated service treatments according to the requirements of each service. It becomes clear that the future networking control is expected to take on a much broader meaning than just relating to routing of connections based on simple database look-ups.

It is also claimed that decentralization approaches, supporting redundant connectivity and multihoming with automated control capabilities, are more promising for scalability, robustness, and availability than centralized solutions, especially in large scale network environment [39], [40], [41], [42]. Indeed, the explosive expectations from the ITs are making central controllers bottlenecked. As stated in [43], distributed control efforts focus on mechanisms for enabling networks with self-awareness, self-optimization, and self-management capabilities, whereby network elements can adapt themselves to contextual changes without any external intervention. Thus, decentralization allows the support of simultaneous operations at different entities throughout a network, seeking better performance with less resources (e.g., bandwidth, Central Processing Unit - CPU, memory and energy) consumption. However, it requires the synchronization of control information among the distributed entities to avoid wrong and incompatible decisions [44] while exceeding signalling (e.g., as in P2P networks [45] and Ad hoc networks [46]) consumes too much resources. Moreover, the lack of appropriate solutions for decentralization is forcing major designs to centralization (e.g., Enthrone [47], EuQoS [48]), or each distributed system to deploy its own strategies in the form of overlay; the complexity of the Internet continues to increase even more by the addition of new protocols and mechanisms atop of the current layers [45]. Considering the inexistence of appropriate analysis of the pros and cons of each of the two approaches in the literature, Song et al. in [49], provide results demonstrating that distributed system is preferable in large network or when traffic load is high and uniformly distributed. Centralized system performs better in simpler scenarios or when the majority of calls are initiated or destined to one edge node. It is therefore apparent that decentralization will play a key role in the current and future networks as they start to extend service to human-surrounded objects (e.g., Internet of the things).

Bearing the aforementioned issues and challenges in mind, the research community, in both academia and industry, recognizes the increasing limitations of the current Internet in terms of network management which is difficult to deploy, has security concerns, and the “best effort forwarding” has failed to meet requirements for QoS and added-value applications. It is broadly studied that the Internet architecture design strongly needs reconsideration and many proposals [50], [51] including “clean slate” approach [52] were made available. Efforts in OpenFlow [53] and

GENI [54] attempt to encourage networking vendors for programmable switches and routers (e.g., using virtualization) that can process packets for multiple isolated experimental networks simultaneously. The main objective is to exploit practical and sufficiently realistic environments (e.g., real traffic at large scale) to allow for experimenting innovative ideas to gain the confidence needed for widespread deployment of new approaches including alternatives to IP. In GENI, nationwide research facility was suggested for experimentation where a researcher will be allocated a slice of resources (e.g., links, packet processing elements, end-hosts) across the whole network so that researchers can program their slices to behave as they wish. The OpenFlow proposes to circumvent costly settings by envisaging that vendors add OpenFlow protocol to the equipments in college campus. The idea relies on flow-tables (available in most of modern Ethernet switches and routers) which includes packet header field that defines flows, and specifies how the packets of a flow should be processed. Usually, it runs at line-rate to implement firewalls, Network Address Translation (NAT), QoS, and to collect statistics of each flow [53]. Hence, the open protocol is introduced to program the flow-table in different devices in a way that network administrator can partition traffic into production and research flows where the latter could be exploited by researchers. There is no longer doubt that significant efforts were still necessary to evolve the networking control technologies.

From system survivability perspectives, unpredictable failures (e.g., of links and nodes), usually caused by natural disasters (e.g., fire, earthquake, etc.), malicious attacks, hardware faults, and human mistakes, threaten network normal operations. The term survivability refers to the ability of a network to assure service continuity to a certain degree in the presence of these challenges [55]. The survivability approaches are generally classified into protection-based and restoration-based [55], [56]. The main objective is to achieve service stability through minimum recovery time while assuring differentiated control and maintaining maximum resource utilization at low cost [57]. It has been one of the fundamental design goals, in all existing networking technologies (e.g., IP, ATM, MPLS, etc.) [55], to provide stable operations regardless of the scale, the magnitude, the duration and the type of failures [58], [59]. As an example in IP infrastructures, the widely deployed routing protocols (e.g., Open Shortest Path First - OSPF [4], Intermediate System to Intermediate System - IS-IS [60]) are able to reestablish connectivity after the failure of network elements. Hence, survivability control must be an integral part of any new design for the current or future networking scenarios.

1.2 *Objectives and Main Contributions*

The main objectives of this Thesis consist in investigating scalable QoS architectures and network control mechanisms to allow for optimizing networking control overall performance. For

this purpose, we steered our focus on the trade-off between scalability in terms of control signalling rate/frequency minimization and the waste of resources confronted in aggregate resource over-reservation approaches due to dynamic characteristics of network environments such as in distributed scenarios. Our goal is not to propose a "clean slate" design, but a pragmatic and compatible technology with existing standards to flexibly advance the state-of-the-art in the area of networking control.

The overall control designs proposed in this Thesis are targeted at a single networking domain (e.g., area or Autonomous System - AS) with well-defined boundary (similarly to DiffServ or MPLS domains) composed of edge nodes (ingress and egress nodes) through which traffic may enter or exit the domain respectively, and core nodes are placed inside the domain. This assumes that each network domain can deploy its own control technology and inter-domain connections (e.g., between different administrative domains) can be assured through any specific approach such as negotiable Service Level Agreements (SLAs) and Service Level Specifications (SLSs) [47].

Hence, we propose new QoS and networking control mechanisms for centralization and decentralization designs with support for survivability, using aggregate resource overprovisioning concept. In our centralized design, a central server implements a CDP entity, while every edge node embeds a CDP in the decentralized approach. The internal architecture of a CDP depends on the design model, that is, whether it is centralized or decentralized, as we will detail later in the Thesis. A CDP is therefore the responsible for maintaining a good knowledge of the underlying network topology and related resource conditions in real-time manner for making policy and control decisions based on accurate information inside the network. To achieve this in the decentralized approach, all available CDPs cooperate as a means to dynamically exchange appropriate control information for synchronization to changes in network topology and related resource states. The decisions taken by a CDP are translated into commands and conveyed in signalling messages to the core nodes which host a Decision Enforcement Point (DEP) entity for the enforcements while being kept simpler.

Besides, every edge node implements the DEP entity with additional key functions (e.g., gate control and traffic conditioning), which are usually pushed to network border. Further, a CDP builds multiple QoS-aware edge-to-edges multicast trees and dynamically manages aggregate bandwidth over-reservation among the CoSs configured on the trees, aiming to establish sessions without per-flow signalling for QoS or synchronization among distributed CDPs. The use of multicast trees is a means to ensure that the packets that belong to a flow mapped to a tree are pinned to the tree so they enjoy the QoS destined to them. Therefore, in the description of our designs in this Thesis, the terms path and tree are interchangeable. Moreover, the terms edge

node/router and border node/router are interchangeable. The main contributions of this Thesis are summarized in the following.

Scalable QoS and Architecture for Centralization of Network Control

We propose a new centralized control architecture design called ACA. In ACA, a central server, embedding a CDP, creates all possible edge-to-edges trees inside the network under its control, and records key control information of every outgoing interface on the trees (e.g., interface capacity, interface ID and CoSs configured on the interface). Any session request to the network is sent to the CDP which maps authorized requests to appropriate CoS and best trees through admission process (depending on requested QoS and network current conditions) in a way that balances traffic load across the domain. Upon granting access for a new session, releasing or readjusting an ongoing session requirements in a CoS on a tree, the CDP automatically updates, in its local database, the resource utilization parameters of the CoS for every outgoing interface that belongs to the tree. This way, the CDP maintains a good knowledge of the whole network topology, the existing trees and the related link resource usage statistics in real-time manner, which provides support for network control sub-systems such as QoS, traffic engineering, network planning and mobility. This reduces the need for frequent trees' probing which is very important to improve scalability. Moreover, such real-time view on resources (bandwidth) statistics of underlying network is a key requirement to efficiently support aggregate resource over-reservation for addressing the trade-off between QoS signalling overhead minimization, QoS violation, waste of resources, and therefore, unnecessary increase of session blocking probability.

Based on the hereinabove architectural support of the ACA, we propose two aggregate resource over-reservation control algorithms, the COR and the ECOR. COR and ECOR allow for dynamically defining over-reservation (surplus of reservation) parameters for CoSs configured on each outgoing interface on a tree, based on the current resource conditions of the interface to avoid per-flow QoS signalling. More importantly, the computational functions of both COR and ECOR prevent CoS starvation and waste of resources, while the signalling overhead is significantly reduced. While COR avoids over-reserving too much resources for each CoS, the ECOR allows as much over-reservation as possible and thus, the latter enables for optimizing the minimization of signalling and related processing overhead. We implement these algorithms in the ACA and in our decentralized designs described below, and study their performance metrics.

Scalable QoS and Architecture for Decentralization of Network Control

We propose a generic mechanism for decentralization of network control called ACOR. ACOR enables multiple CDPs distributed at network border to cooperate to exchange communication trees and related resource usage information, such that each CDP is able to maintain a good knowledge

of the network topology (e.g., nodes and links) and the related links resources statistics in real-time manner. From scalability perspective, ACOR cooperation is selective, meaning that information is exchanged between only the CDPs which are concerned and unnecessary broadcasting is avoided dynamically. Moreover, we propose a VOPR concept which allocates a share of aggregate over-reservation of each interface to each of the trees that use the interface. As a result, each CDP is enabled to process several service requests on a tree without requiring synchronization, as long as the VOPR of the tree is not exhausted, and thus, the ACOR synchronization signalling rate is also kept low. Further, we exploit the ACOR support for aggregate resource over-reservation techniques, and implement the COR and ECOR algorithms to provide scalable resource and admission control functions with low QoS reservation signalling and the related overhead without wasting resources. Moreover, we propose a Next Steps In Signalling (NSIS) compliant control signalling protocol, called ACOR-P, which defines appropriate message structures, fields and objects in support for all the control mechanisms proposed in this Thesis.

E-ACOR and Architecture for Decentralization of Network Control

While ACOR allows for optimizing QoS reservation signalling overhead, its synchronization signalling rate increases rapidly with the increase of the number of trees that use bottleneck interfaces inside a network. Moreover, as the VOPR exhausts upon every session request during network congestion period of time, it forces ACOR to per-request synchronization in congestion situations. Therefore, we propose the E-ACOR. In particular, E-ACOR extends the VOPR concept by aggregately allocating the over-reservation of an interface to all the trees that originate at the same CDP. This way, each CDP may process more session requests on a tree without requiring synchronization when compared with ACOR using the VOPR per tree, since session demands are mostly unpredictable. Moreover, E-ACOR enables each CDP to efficiently track network congestion information without undue control signalling overhead in a way to prevent unnecessary synchronization signalling when network is congested. More importantly, E-ACOR allows for optimizing the synchronization overhead reduction while keeping the optimization capabilities of ACOR in terms of QoS reservation signalling overhead reduction without QoS violation or waste of resources.

Advanced Control for Network Survivability

We propose the SACOR in support for stable operations and service continuity in ACOR in case of failures “Down” (e.g., links/nodes failures) or when previous failures recover “Up”. In particular, core nodes mainly detect and announce failure or recovery events to all CDPs using our proposed flooding-based approach. Hence, upon receipt of failure or recovery notification(s), the CDPs are enabled to cooperate to quickly adapt to the changes imposed in terms of topology, links

resource conditions, and timely re-routing of traffic flows. In other words, SACOR pushes survivability control load and complexity to CDPs, and core nodes are left simpler for fast convergence and scalability purposes. Regarding differentiation of flows re-routing, we propose VOPR-based Re-routing (VR) and Preemptive-based Re-routing (PR) techniques, which allow for fast traffic switchover upon failures without requiring ACOR synchronization or resource reservation signalling. The VR re-routes flows based on available VOPRs, and the PR preempts lower priority flows to accommodate higher priority ones. Further, Available Reservation-based Re-routing (ARR) and Reservation Readjustment-based Re-routing (RRR) schemes are introduced for re-routing remained flows after the VR's and PR's operations. The ARR re-routes traffic after the CDPs' synchronization to overall changes occurred in network resource utilization statistics, and the RRR enables for readjusting reservations parameters on trees upon need to avoid dropping traffic unnecessarily or wasting resources upon failures. Our simulation results show that SACOR effectively provides differentiated survivability under fast convergence operations while efficiently using the network resources.

1.3 *Publications*

The contributions of this Thesis work resulted in the following number of publications.

1.3.1 Pending Patents

- Evariste Logota, Sargento Sargento, Augusto Neto, "Um Método para Controlo Avançado de Sobre-Reservas Baseado em Class de Serviço e Sistema para a sua Execução (A Method and Apparatus for Advanced Class-based Bandwidth Over-reservation Control)", 105305, September, 2010.
- Evariste Logota, Sargento Sargento, Augusto Neto, "A Method and Apparatus for Class-Based Networks Control", CI-12-029, January, 2013.

1.3.2 Pending Journals

- Evariste Logota, Carlos Campus, Susana Sargento, Augusto Neto, "Advanced Multicast Class-based Bandwidth Over-Provisioning", Elsevier Computer Networks Journal, 2012 (submitted in July 2012).
- Evariste Logota, Carlos Campus, Susana Sargento, Augusto Neto, "SACOR: Survivable Advanced Class-based resource Over-Reservation", Elsevier Computer Networks Journal, 2012 (submitted in November 2012).

1.3.3 Pending Conference

- Evariste Logota, Carlos Campus, Susana Sargento, Augusto Neto, "Scalable Resource and Admission Management in Class-based Networks", (submitted to the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Communications (ICC 2013)).

1.3.4 International Proceedings with Independent Reviewers

- Evariste Logota, Augusto Neto, Susana Sargento, "COR: an Efficient Class-based Resource Over-pRovisioning Mechanism for Future Networks", IEEE Symposium on Computers and Communications (ISCC 2010), Riccione, Italy, June 2010.
- Evariste Logota, Augusto Neto, Susana Sargento, "A New Strategy for Efficient Decentralized Network Control", IEEE Global Telecommunications Conference, (IEEE GLOBECOM 2010), Miami, Florida (USA), December 2010.
- Augusto Neto, S. Sargento, Evariste Logota, J. Antoniou, F.C Pinto, "Multiparty Session and Network Resource Control in the Context Casting (C-CAST) project", Second International Workshop on Future Multimedia Networking (FMN 2009), Coimbra, Portugal, June 2009.
- Augusto Neto, S. Sargento, F. C. Pinto, Evariste Logota, "Context-Aware Session and Network Control in Future Internet", IEEE International Conference on Communications (ICC 2009), Dresden, Germany, June 2009.

1.4 *Thesis Organization*

This section introduces the structure of the Thesis as in the following.

Chapter 2 provides an overview of the more relevant work within the scope of this Thesis in a way that facilitates the understanding of the state of the art of the research aspects addressed in the Thesis. In particular, it presents the major IETF standardized QoS and resources control architectures such as the InServ, DiffServ and MPLS. Moreover, it describes the existing admission control models (the active measurement-based, the passive measurement based, and the parameter-based), and introduces IP multicast technology. Besides control signalling protocols, the architecture and requirements of NGNs have also been explored together with resources and admission control standards, and the related key building functional blocks. Further, it introduces networking control models with focus on centralization and decentralization approaches, including scalable resources and admission control proposals along with key control frameworks. Relevant proposals and mechanisms for network survivability, and context-awareness have also been surveyed in this chapter.

Chapter 3 introduces our novel resource over-reservation algorithms (COR and ECOR). It also describes our ACA architecture which deploys a single CDP as the responsible for maintaining global network topology and the related links resource statistics to assure the overall network control. ACA integrates the COR and ECOR, and demonstrates superiority over state-of-the-art solution by drastically reducing control signalling, and therefore, the related processing overhead. This chapter provides also a generic-purpose analytical model, which allows for assessing key control parameters that generally challenge resource over-reservation performance in terms of signalling minimization and waste of resources.

Chapter 4 describes the self-organizing multiple edge nodes decentralization control mechanism, called ACOR. It includes an introduction to highlight the driving motivation behind the work. ACOR provides a generic-purpose protocol for decentralization control, and supports network control sub-systems through a good knowledge of network topology and the related links resources statistics, which are obtained by keeping low signalling and related overhead. This chapter provides both analytical and simulation studies of the proposed approach. It also describes our ACOR-P control signalling protocol, an NSIS compliant protocol, proposed to support our centralization and decentralization control mechanisms in this Thesis.

Chapter 5 describes the E-ACOR decentralization control mechanism. E-ACOR introduces an aggregate VOPR concept which consists in aggregating the fine-grained VOPR approach of ACOR. Moreover, E-ACOR tracks network congestion information efficiently to allow for optimizing the synchronization signalling overhead reduction. As a result, E-ACOR allows for optimizing overall control signalling (QoS reservation and synchronization between CDPs) overhead in decentralized network environment. Finally, the chapter includes analytical and simulation evaluation for comparison between the E-ACOR and ACOR.

Chapter 6 describes the survivability mechanism of SACOR proposed in support for stable operations of ACOR (e.g., fast convergence and differentiation of flows re-routing) in the face of failures or recoveries. The proposed functions also apply to ACA and to the E-ACOR architectures.

Chapter 7 introduces the final conclusions and the main contributions of the Thesis. It also highlights possible directions for further developments of the research aspects addressed in this Thesis.

Chapter 2

Related Work

The networking transport performance objectives are generally achieved from traffic-oriented perspective in terms of QoS requirements (e.g., bandwidth, delay, etc.), and resource-oriented perspective in terms of network resource management taking into account key features such as efficiency, scalability, cost-effectiveness, etc. From the beginning of the NGNs study, resource and admission control functions and protocols have been considered and intensively discussed as one of the main functionalities for transport service (unicast or multicast) provisioning with acceptable quality and cost [10]. This chapter introduces general background of network architectural control as well as resource and admission control in the scope of the NGNs as being the focus of this Thesis.

This chapter is organized as follows. Section 2.1 presents the major IETF frameworks for QoS control and section 2.2 describes the main existing admission control models. Section 2.3 introduces IP multicast technology while section 2.4 focuses on control signalling protocols. The section 2.5 presents an overview of NGNs architecture and requirements together with resources and admission control standards, including related key building blocks. Moreover, the section 2.6 introduces network control models with centralization and decentralization approaches, and section 2.7 describes scalable resources and admission control proposals along with key frameworks. Further, section 2.8 introduces proposals and mechanisms for network survivability, and section 2.9 addresses context-awareness. Finally, section 2.10 concludes the chapter.

2.1 *Major Frameworks for QoS Control*

In order to provide QoS support over the originally best-effort-based IETF standardized three major frameworks: (1) IntServ [2]; (2) DiffServ [3], and (3) MPLS [6]. Unlike IntServ, which was

designed to provide strick QoS for each individual flow admitted over the Internet, DiffServ was introduced as an extremely scalable QoS architecture. Besides, MPLS was introduced not only to integrate QoS and traffic engineering features, but also it achieves fast routing based on labels.

2.1.1 Integrated Services

In order to provide end-to-end QoS for each service over the best-effort-based Internet, the IETF developed a fine-grained QoS control architecture, the IntServ [2]. IntServ model supports two types of services: (1) Guaranteed Service (GS) [61], targeting hard real-time applications, which can mathematically guarantee bandwidth, delay and jitter; (2) Controlled-Load (CL) service [62], which provides soft guarantees to applications that can adapt to network conditions within a certain performance window. The QoS guarantee of a flow is obtained through the reservation of adequate resources at every node that happens to lie on the communication path taken by the flow from its source to its destination, usually resorting to the *Resource Reservation Protocol* (RSVP) [63]. To this end, a user's application, the traffic source, must identify its flows, using the 5-tuple (source IP address, destination IP address, source port, destination port, transport protocol), and specify the flows characteristics and the service requirements in terms of traffic envelope and the amount of bandwidth to be reserved for each flow.

Hence, the source/sender side encapsulates the traffic characteristics using a container called *Traffic Specifications (*TSpec*)* in an RSVP specific message, the *PATH* message, and sends it to the traffic receiver. Upon receiving the message, the receiver specifies the desired QoS as a *Reservation Specification* (RSpec) object in a specific message called *RESV* message and sends it to the source. Hence, every node on a path must interpret the RSVP messages and perform admission control based on the required QoS and its available resources. In case the admission control succeeds, the node retrieves the QoS required from the message and enforces the reservations on its corresponding interface(s), by configuring the scheduler on the interface(s) [64], [65]. Also, the node records the traffic identification parameters and properly configures its routing tables. As illustrated in Figure 2.1, every node on a path must be RSVP-aware not only to reserve QoS, but also to maintain states for each flow so as to assure an end-to-end QoS control for each flow.

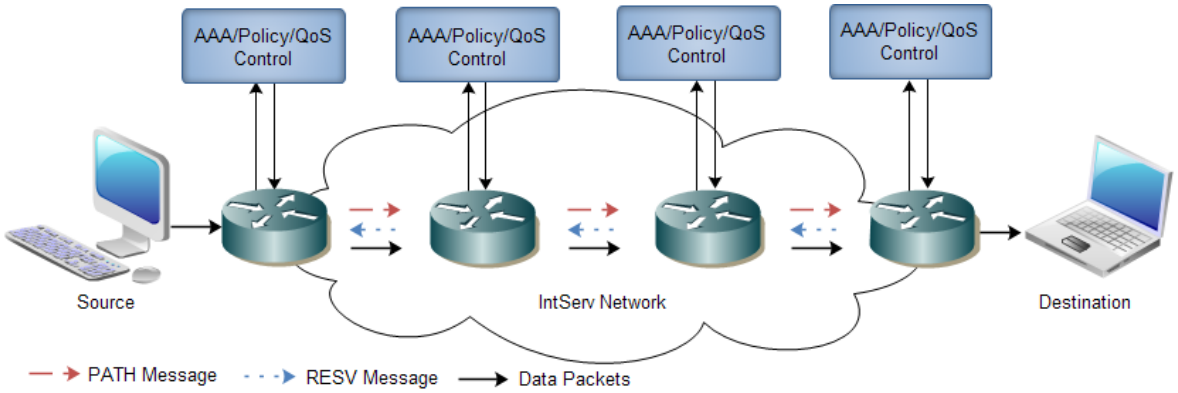


Figure 2.1. Illustration of IntServ aware network scenario.

As a consequence, such per-flow fine-grained states and signalling operations in IntServ lack scalability due to excessive control overhead and have been severely criticized [20]. Therefore, IETF introduced the DiffServ as an alternative QoS architecture standard for the Internet.

2.1.2 Differentiated Services

The DiffServ [3] has been introduced by the IETF as a coarse-grained, a class-based alternative mechanism to the IntServ paradigm. It entends to cope with key shortcomings (scalability and flexibility) of the IntServ while keeping QoS support over the Internet. Different from the end-to-end IntServ approach, a Differentiated Service (DS) domain has a well-defined boundary composed of ingress and egress nodes through which traffic may enter or exit a DS domain, respectively, while core nodes are placed inside the domain, as in Figure 2.2. There are two types of DS domains: (1) stub domains in which the nodes are typically endpoints in a network flow — network traffic either originates at or is destined for a node in a stub domain (a router at a local Internet Service Provider - ISP); (2) transit domains in which the nodes are typically intermediate in a network flow — traffic usually pass through it (backbone routers). Two important concepts such as SLA and Traffic Conditioning Agreement (TCA) have been defined in DiffServ architecture. An SLA defines a contract between a provider and a customer in terms of the specifications of the service(s) to be provided, where a SLS includes the traffic treatment and performance metrics. A customer may be a single user, a user organization source domain or another DS domain (upstream domain). In addition, a TCA may be included in an SLA to specify the traffic conditioning rules, that is, the rules for packet classification, the traffic profile and the corresponding rules for metering, marking, shaping or dropping.

All traffics entering a DS domain are classified into a limited number of CoSs (a.k.a., Aggregates behavior) supported inside the domain such that traffic are treated aggregately and not per-flow. Packet classification may base on multi-field in packets header (e.g., source address, destination address, source port, destination port, protocol type). An aggregate behavior or a CoS of

a packet is identified by a single 6 bits Differentiated Service Code Point (DSCP) [66], thus allowing traffics to be treated aggregately, and not per-flow. Hence, based on a domain's service provisioning policy, a Per-Hop Behavior (PHB), consisting of a means by which a node allocates its resources to a particular CoS on its interfaces, is implemented by employing some buffer management (e.g., Random Early Detection - RED) and packet scheduling mechanisms (e.g., Weighted Fair Queuing - WFQ). These per-hop behaviors are required in network nodes to assure differentiated treatment of packets along the communication paths.

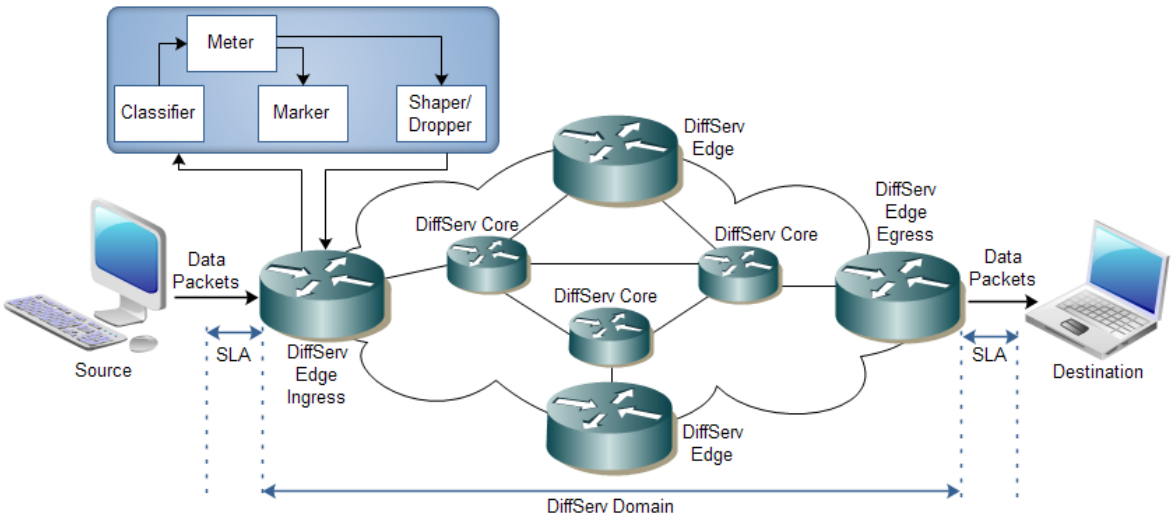


Figure 2.2. Illustration of DiffServ aware network scenario.

In addition to Best-Effort (BE), DiffServ defines the Assured Forwarding (AF) [67] and Expedited Forwarding (EF) [68] PHB groups. The EF service is used to build a low loss, low latency, low jitter, assured capacity with peak rate allocation ("Virtual Leased Line" or Premium service), achieved by using priority queuing, along with a rate limiting on the CoS. The AF service provides relative guarantees, assuring high probability of forwarding conformant packets, meaning that some packets may be dropped more aggressively during congestion period of time.

However, packet classification and prioritization are implemented for different service levels [69] and control degradation can worsen in dynamics re-routing situations [70] since traffic flows may change their routes without taking into account the available resource conditions in the new paths. Moreover, high priority traffics can deprive lower priority services of resource. Pengxuan et al., [71] argue that it is very difficult to guarantee differentiated QoS in multiple distributed edge nodes environment. As the core nodes are required to be simpler, it is easy that they get overwhelmed when edge nodes inject traffic packets on them in distributed manner and there is no synchronization between the edge nodes. They combine the functionalities of DiffServ and MPLS (Multiprotocol Label Switching) to alleviate the impact of this probem on QoS performance. In particular, edge nodes implement DiffServ functions mainly for marking and dropping packets, and

exploit MPLS Traffic Engineering functions to dynamically re-route/remap traffic flows to reduce the congestion problem. The core routers are mainly responsible to forward the packets which are scheduled according to the PHB of each packet as defined by edge node using DiffServ.

Considering the scalability features in DiffServ and the QoS guarantee capabilities assured in IntServ through end-to-end admission control and dedicated resource allocation strategy [72], RSVP has been extended to support scalability by means of aggregate reservations [22]. The protocol has also evolved to support features such as security [73], MPLS [74] and GMPLS [75].

2.1.3 MPLS and MPLS-based Approaches

The MPLS is a connection-oriented packet-switching technology which inherently supports traffic engineering by using explicit paths and offers a great deal of flexibility to route traffic around link failures, congestion and bottlenecks. As illustrated in Figure 2.3, an MPLS *Label Switching Domain* (LSD) consists of well-defined boundaries, encompassing entry and exit nodes called *Label Edge Routers* (LERs), and interior nodes called *Label Switch Routers* (LSRs). In MPLS, the Constraint-based Routing Label Distribution Protocol (CR-LDP) [76] and RSVP are well-known two signalling approaches to manage label space to provide QoS reservation and traffic engineering. In particular, the CR-LDP takes various metrics (e.g., link bandwidth, delay, hop count, etc.) into account to provide explicit routes based on QoS and CoS requirements, while the RSVP-Traffic Engineering (RSVP-TE) [74] allows for QoS reservations along the paths upon need. Note therefore that *Label Switched Paths* (LSPs) may be established without bandwidth reservation, unless bandwidth requirements for the LSP are signalled at LSP establishment time [77].

In order to assure a proper transport of services along a given LSP, all LSRs on the LSP must agree in advance on the meaning of the labels to be used to forward traffic flows between and through them along the LSP. This means that an LSP must be established between LERs and LSRs prior to any data forwarding, which is achieved through the Label Distribution Protocol (LDP). This way, LSPs are protocol agnostic (e.g., independent on a particular layer-2 technology), while they permit quasi circuit switching capabilities for aggregate traffic transport. In order to push control complexity to the border, the LERs are responsible for defining the explicit LSPs throughout a network. Moreover, they potentially maintain and leverage the knowledge of the underlying network resource capabilities and a set of pre-defined control policies to select less congested LSPs to map traffic flows, thus balancing traffic load across the network.

Furthermore, the LERs are responsible for classifying every ingress packet into a three bits Forwarding Equivalence Class (FEC). Basically, an FEC describes how aggregate traffic flows are forwarded (the treatment) along the LSPs. After that, each classified packet is mapped to an LSP

through encapsulation with appropriate label pushed atop its header and then forwarded to the next LSR on the selected LSP. Upon receiving a packet, an LSR looks up the label and re-labels it according to the corresponding FEC [78] and the packet is forwarded to the next LSR. This encapsulation and forwarding process is repeated hop-by-hop until the last router pops the shim header to deliver the packet to the endpoint or to the next router in subsequent domain based on the IP header of the packet.

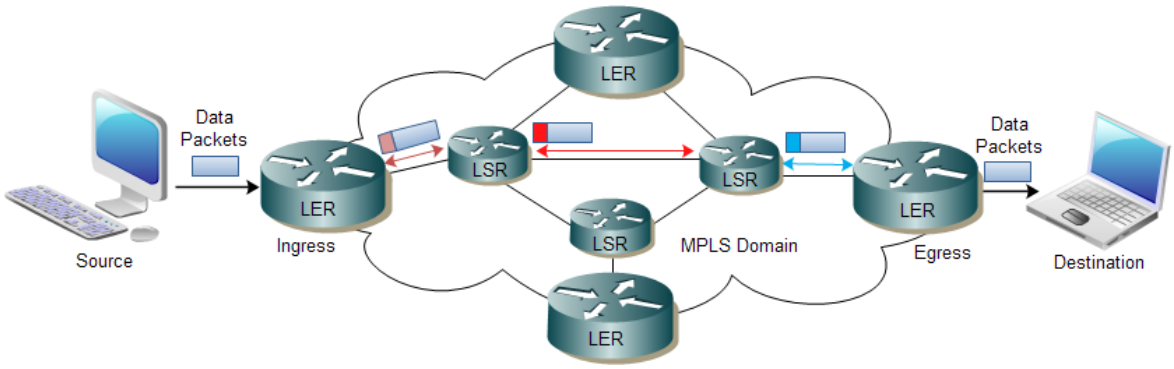


Figure 2.3. Illustration of MPLS enabled network scenario.

Moreover, the RSVP has been extended to support multicast, especially the Traffic Engineered (TE) point-to-multipoint (P2MP) LSPs in MPLS and GMPLS networks [79], where the solution relies on RSVP-TE and not on the legacy multicast routing protocol in the Service Provider core. Although the potential of deterministic LSPs and the pertained control concepts are fundamental for Traffic Engineering, the dynamic label handling (encapsulation) is a major problem in MPLS. This means that, besides the forwarding function, all routers within an LSD must examine every incoming packet label and decide whether the packet should be encapsulated by adding a new label to the topmost of the label stack of the packet, swapped by another label or removed. Moreover, while constraint-based routing increases network utilization, it adds more complexity to routing calculations, since the path selected must satisfy QoS requirements. Also, the excessive control signalling is another crucial shortcoming, since QoS reservations and LSP paths setup signalling frequency increases with the number of session connection requests.

As discussed in [9], key features such as scalability, service convergence, QoS and cost-efficiency are essential requirements for the emerging network transport technologies to cope with the current strong need of transforming operators' network infrastructures with reduced capital and operational expenditures in support for the growing demands of packet-based services. This has been a driving force to the birth of MPLS-TP [8] which is being defined in IETF (groups – MPLS, Pseudo Wire Emulation Edge-to-Edge architecture - PWE3, and Common Control and Measurement Plane - CCAMP) and the ITU-T SG15 as a simplified and enhanced version of MPLS. In particular, MPLS-TP turns off some of the MPLS functions such as Penultimate Hop

Popping (outermost label of a packet is removed by a LSR before the adjacent LER), LSPs merge, and Equal Cost Multi Path, and adds a few enhancements, mainly in the area of Operation, Administration, and Management [8], [80]. Hence, it enhances the protocols and mechanisms that are used to set up the LSPs, and those (e.g., Framing, forwarding, encapsulation and resilience) that are used to forward the data packets. Moreover, while MPLS supports a robust and mature dynamic control plane with protocols such as Open Shortest Path First-Traffic Enginering (OSPF-TE) [81], Intermediate System to Intermediate System-Traffic Engineering (IS-IS-TE) [82], RSVP [74], LDP [83], and Border Gateway Protocol (BGP) [84], dynamic control plane is optional in MPLS-TP, knowing that GMPLS and Label Distribution Protocol [83] can be used to set up LSPs and pseudowires respectively in the context of MPLS-TP [80].

In [23], a bandwidth management system is demonstrated in multiple edge nodes environment for assuring QoS using a combination of IP flow control at network edge and MPLS DiffServ-TE (control of traffic on per CoS basis and not on per-flow basis over LSP). The authors show that Call Admission Control (CAC) and traffic control can be implemented on edge nodes to allow for preventing excessive QoS reservation signalling to core nodes inside a network control domain. To achieve this, they studied that the CAC at each edge node requires a proper bandwidth management system to take into account the dynamically shared resource utilization statistics of the core links which are shared by the LSPs originated by all edge nodes. Hence, they deploy a central network management system (NMS) which manages the LSPs, IP flows and the bandwidth to be used by CAC of the edge nodes routers. The NMS is connected and collects this vital information from Management Information Base (MIB) available on network elements via the Simple Network Management Protocol (SNMP) and IP flow via Netflow [85]. While this demonstrates key issue and challenges to be addressed in the modern networking to improve performance, the proposed solution is centralized and confronts serious scalability problems.

As being class-based architectures, the approaches proposed in this Thesis can be deployed in DiffServ, MPLS and the MPLS derivatives enabled scenarios.

2.2 *Admission Control Models*

Network Resource Admission Control techniques, consisting of accepting or denying service requests are generally classified as active measurement-based, passive measurement-based, and parameter-based.

2.2.1 Active Measurement-based Admission Control

Also known as *probing-based admission control*, the AMAC consists of probing communication paths, usually based on a packet sequence or traffic with the same characteristics as

those of the service which is waiting for admission decision [15], [16]. This usually provides available bandwidth, delay or packet loss information about candidate paths to assist admission decisions. This approach, in essence, does not require resource reservation for services along paths. However, it imposes long session setup time while signalling overhead is of major concerns, since paths must be probed before admission decisions. Moreover, the existing probing techniques mostly suffer from complexity, accuracy issues, and only provide soft QoS guarantees [17], [15], [19]. Studies show that probing may lead to "*Trashing Regime*" [18] in multiple distributed ingress nodes environment. This means that multiple simultaneous probing traffic easily overload network especially during congestion time.

2.2.2 **Passive Measurement-based Admission Control**

In Passive Measurement-based Admission Control (PMAC) control models, each node on a path is expected to measure and be aware of the average real traffic data load on each of its interfaces, so as to obtain the available bandwidth based on the capacity of the interface [86]. In class-based networks, each node must measure the user's real traffic load in each CoS on each of its interfaces, so as to deduce the available resource in each CoS based on the maximum allowable traffic load per CoS (maximum allocated bandwidth per CoS) [16]. Hence, upon receiving a service request, a network Control Decision Point usually signals the candidate path(s) along which each node may accept or deny the request according to its available resources. Besides the signalling overhead, computational overhead is another concern in this approach. Moreover, it mostly applies in the context of soft QoS services due to performance degradation, which is subject to dynamic traffic characteristics and the fine-tuning of measurement design model's parameters on nodes across a network [87], [88].

2.2.3 **Parameter-based Admission Control**

As alternative to the previous two approaches, the PAC considers the amount of resources already granted to currently running services (sum of bandwidths granted) and the total allowable capacity to obtain available resource (capacity minus sum of granted bandwidth) to assist admission decision. As a result, this model is simple and suitable to deliver various types of services including Premium services or hard QoS services (e.g., Guaranteed Service and Expedited Service) without performance degradation [89], [90]. More importantly, this allows for providing each user with what he/she has bought from the service or network provider regardless of whether the user consumes its resource or not. Note therefore that one can opportunistically use the unused resource of users in order to increase revenue, which is a control policy issue.

As one could see, the studies in this section reveal that, the PAC provides better support for QoS guarantees with less complexity. Moreover, it prevents undue signalling load and inaccuracy drawbacks suffered in AMAC and in PMAC). Therefore, the PAC is adopted in this Thesis.

2.3 *IP Multicast Technology*

In late 1980s, Dr. Steve Deering first suggested IP Multicast [26], which consists of using special network addressing [91] to send one single traffic stream to any number of recipients in a group, called a multicast group. As illustrated in Figure 2.4, a single multicast stream sent to an interested group of destinations (destination No. 1 through 4) replicates at branching nodes, and thus, offers efficient network resource utilization in contrast to unicast stream, which unnecessarily sends on shared links as many copies of the same stream as there are users (destination No. 5 through 8) expecting to receive it.

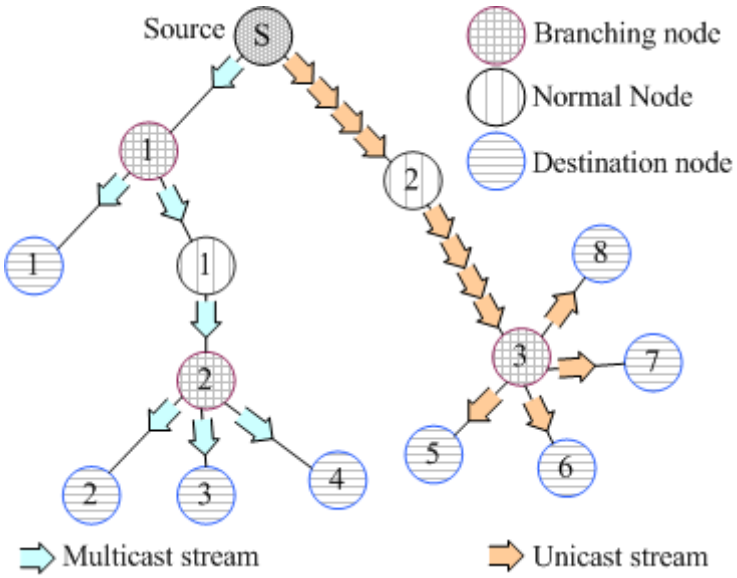


Figure 2.4. Illustration of multicasting scenario.

However, IP Multicast was used only rarely as a communication tool between IP routers and switches until it was approved as the best promising technology for group communications such as multimedia sharing, collaboration between people, social networking (e.g., You Tube [92], Bebo [93], Facebook [94]) over the Internet. Other examples of applications include Internet TV [95], Massive Multiplayer Online Games [96], file sharing, software updates, video conferencing, etc. When using Internet Standard Multicast (ISM) multicast model, a receiver does not need to know the identity of the media source. To receive a datagram destined to a particular group, an upper-layer protocol invokes the IP module to join that group on a specific interface (e.g., JoinHostGroup (group-address, interface)). Likewise, a user may simply leave a group by issuing a LeaveHostGroup (group-address, interface). As such, a node may join the same group on more

than one interface, or more than one upper-layer protocol can join the same group. However, this raises media source filtering and security concerns in multicast delivery. Hence, the multicast has evolved to the Source Specific Multicast (SSM) model [35]. With SSM, a receiver must know a multicast channel (S, G) where S is the media source address and G is the Group address, and explicitly join the channel [97] to be able to receive the data traffic. There also exists a Small Group Multicast (SGM) model which allows a packet from sender to contain the list of all receivers and may be used when the group is small and the overhead introduced is neglected. In general, a multicast session procedure involves: (1) multicast session/group creation; (2) multicast tree construction; (3) data transmission, and (4) multicast session termination [98] where the multicast protocols, classified into dense and sparse modes, are the responsible for finding multicast paths, managing the multicast group(s), and building the distribution trees which may be source trees or shared trees.

2.3.1 Multicast Routing and QoS Control Protocols

The source multicast tree algorithms use the notion of shortest path tree rooted at the source. Each branch of a tree is the shortest path from the source to each group member and delay may be minimized [99]. However, they pose scalability issues under large number of groups with each group having a large number of sources, since routers storage capability can be stressed. Several existing source-based Multicast routing protocols include Distance Vector Multicast Routing Protocol (DVMRP) [100], Protocol Independent Multicast - Dense-Mode (PIM-DM) [101], and Multicast Open Shortest Path First (MOSPF) [102]. Derived from Routing Information Protocol (RIP), the DVMRP is a broadcast-and-prune style algorithm, meaning that a packet multicast by a source is flooded to all end hosts and those who are not interested send “prune” message up the distribution tree. It keeps track of the return paths to the source (Reverse Path Forwarding - RPF no spanning trees) and builds efficient shortest-path trees from any source. The PIM-DM, similar to DVMRP, also floods multicast datagrams to all multicast routers and uses “prune” messages to prevent future messages from propagating to routers without group receivers. Hence, these protocols scale poorly due to flooding overhead. Besides, the MOSPF is a link state routing protocol which, in addition to its link state advertisement, associates the list of groups for which it has local receivers. Thus, it builds the map of the network topology and selects the best path to the required receivers using Djikstra’s shortest path algorithm and thus, it is limited to link-state protocol capable networks.

Regarding the shared multicast tree algorithms, they introduce a single location in the network called core or Rendezvous Point (RP), and build a single shared tree which spans all the members whose root is the RP node. The sources register to the RP and receivers join sources through the

RP. These algorithms are more scalable and highly suitable for sparse groups, since they drastically minimize protocol overhead and the amount of state information that needs to be maintained at each router. Nevertheless, the sharing does not favour the scenarios which run with multiple high data rate sources due to traffic concentration, while the end-to-end delay is not optimized. Some existing shared tree based multicast protocols include Core Based Tree (CBT) [103] and PIM-Sparse Mode (PIM-SM) [104]. Ken Carlberg et al, proposed a one-to-many algorithm based on shared tree [27]. With these protocols, new join messages instantiate the forwarding state at routers along their way towards the core/rendezvous point. Thus, the CBT (hard state protocol) builds a single bidirectional Shared Tree, while the PIM-SM (soft state protocol) sets up uni-directional shared distribution trees for data transmission from the source(s) to the receivers. In PIM-SM enabled networks, a router with highest IP address is Designated Router (DR) for its subnet, and therefore, is responsible for sending Prune/Join messages to the RP. DR determines the RP for a group using a hash function, while the information about RP is obtained by sending Bootstrap messages. It is worth noting that PIM-SM offers a particular advantage by allowing switching of receiver connectivity from shared tree to source tree [105]. It turns out that shared tree improves scalability, but the tree obtained is not necessarily optimal. Moreover, while the placement and the discovery of the RP pose major problems, the RP is a single point of failure and ISPs are reluctant to depending on RPs run by other ISPs [106].

In order to address the aforementioned issues that hinder scalability in multicast aware networks, several solutions have been proposed. It is suggested that the PIM-SM uses the next hop information provided by the Multiprotocol Border Gateway Protocol (MBGP) [107], and to build the inter-domain multicast distribution tree, while the Multicast Source Discovery Protocol (MSDP) [108] is used to disseminate (by flooding) source information of one domain to other domains. Hence, the interested users in a domain can receive data multicast by sources, and even switch to a shortest path tree when needed. However, MSDP peers exchange messages using Transmission Control Protocol (TCP) connections and RPF-flooding methods, and thus confront scalability problems; other solutions such as the Border Gateway Multicast Protocol (BGMP) [109]/Multicast Address Set Claim (MASC) [110], EXPRESS multicast [111], Simple Multicast [105] have emerged. Generally implemented at edge router(s) of AS domains, the BGMP builds inter-domain bidirectional shared trees rooted at a single AS domain and allows any multicast routing protocol to be used within the domains. The address allocation required for the root located at the domain is performed using MASC. The EXPRESS protocol [111], which has evolved to the well-known PIM-SSM today [35], proposes a single-source service and supports large-scale single source applications (SSM model) such as Internet TV where any interested receiver must join/leave an (S,G) channel.

However, as they have their roots in the best effort based Internet technology, IP Multicast inherently has no support for QoS sensitive sessions. In this sense, many proposals [27], [28], [29], [30] have focused on network resource provisioning with the objectives of maintaining sessions with improved QoS during their entire lifetime. The research papers on QoS multicast mostly concentrate on QoS-constrained multicast routing problem, using per-flow state, and scalability concern were still prevailing [28]. In other words, establishing and maintaining a multicast tree per-group leads to large memory requirement and slow packet forwarding, since a large number of groups implies a large amount of information to be maintained at routers. Moreover, signalling on per-flow basis and packet handling leads to control explosion. Aggregated multicast [36] was then introduced as tree sharing mechanism to reduce multicast forwarding states information, and therefore improve multicast scalability within a transit domain. These techniques enable multicast flows to be aggregated into one flow at ingress router(s) through packet encapsulation, translation [37], [38] or MPLS techniques, and delivered to egress routers via a single multicast tree (aggregated tree). Jin et al addressed the problem of IP multicast flow aggregation over Wavelength Division Multiplexing (WDM) in order to efficiently utilize light-tree [238]. However, by simply deploying DiffServ-based aggregation, the dynamic addition of new members of the multicast group can negatively affect or even violate the quality of the session of other existing traffic if resources were not explicitly reserved before use. This problem and other limitations in terms of asymmetric routes problem required additional functionalities like admission control, resource reservation. Thus, many QoS multicast aggregation proposals started to incorporate logical and intelligent entities (e.g., tree manager, Multicast Controller, QoS Broker, etc.) that focus on QoS routing, admission control, resource reservation, group-to-tree matching, and policy control as in [31], [32], [33]. An Overlay for Source-Specific Multicast in Asymmetric Routing environments protocol (OSMAR) [34], addresses Asymmetric Routing problem, considering that multicast tree creation is normally triggered from receiver to sender. In particular, OSMAR operations assist tree creation on specific paths by changing the next hop values of the MRIB tables on the path from the source to receivers for requested multicast channel (Source, Group). Then, PIM-SSM will use the information to create tree on desired paths. In other words, OSMAR helps building trees where QoS characteristics of certain paths can be taken into account to create QoS-aware trees for data to follow, which is essential to increase network value as envisaged in this Thesis.

2.4 *Control Signalling Protocols*

Signalling in communication networks is defined as a means for network nodes to exchange information between themselves to establish, maintain, and remove control states or configurations.

With the integration of networks and services on packet-based networks, signalling is used to improve the ability to control the increasing diversity of services offered across the Internet. It is therefore used for many purposes, such as resource and admission control, QoS negotiation control, diagnosing communication paths status, configuring devices, firewall pinholes and NAT bindings.

The Diameter protocol [112] is the ITU-T proposed protocol intended to mainly provide Authentication, Authorization and Accounting (AAA) framework for applications, such as network access or IP mobility including roaming. It is also used for general interaction within resource and admission control decision entities (e.g., Resource and Admission Control Function - RACF), which may reside in different operators' networks.

The SNMP [113] commonly bridges communication between network management stations and the managed elements (e.g., hosts, gateways, terminals, etc). In particular, a network management sub-system can collect vital control information (e.g., interface bandwidth, interface ID, dropped packets statistics, etc.) from MIB available on network elements via the SNMP and IP flow via Netflow [85].

The IETF Resource Allocation Protocol Working Group has defined the Common Open Policy Service (COPS) for support of policy provisioning (COPS-PR) [114] as a scalable protocol that allows policy servers (Policy Decision Points - PDPs) to communicate policy decisions to network devices (Policy Enforcement Points - PEPs) [116] with support for multiple types of policy clients. It is based on a query/response protocol using the reliable TCP such that, one side (client or server) would notice quickly whenever the other side is rebooted (or restarted), and communication is on real-time basis between the PEP and PDP. For instance, when a PEP boots up, it can set up a COPS connection to its Primary PDP, provide the latter with information about itself (e.g., hardware type, software release, etc.) and issue a request for certain configurations. COPS is a well fitted protocol for vertical communication between components in the context of Policy Based Management.

The Session Initiation Protocol (SIP) [117] is an application-layer control protocol which is generally used for creating, modifying, and terminating sessions, such as Internet multimedia conferences, Internet telephone calls, and multimedia distribution. The SIP messages used to create sessions carry session descriptions commonly formatted using Session Description Protocol (SDP) [118], and allow participants to negotiate a set of compatible media types and QoS parameters. The multimedia content (e.g., audio, video, etc.) is exchanged between session participants using appropriate transport protocol (e.g., Real-Time Transport Protocol - RTP). A standard SIP configuration includes the elements such as User Agent, Redirect Server, Proxy Server, Registrar and a Location Service. The User Agent resides in every SIP end station which may be a client to issue SIP requests, or a server to receive requests and generate responses. Besides, a caller uses the Redirect Server during session initiation to determine the address of a called device. A caller's SIP

Proxy server is responsible for routing all SIP messages to another entity (e.g., proxy) closer to the targeted user while SIP Registrar handles the registration by placing the information received (the SIP address and associated IP address of the registering device) into the Location Service for its domain. Hence, a Location Service maintains a database of SIP-address/ IP-address mappings which are used by SIP Redirect or Proxy Server to obtain information about callee's possible locations. The SIP messages are text-based similar to HTTP format, and can either be a request or a response to request, and SIP REGISTER message (for registering a user with a service) and INVITE message (for inviting another user in a session) are the pre-dominant messages used.

Driven by the overwhelming integration of services over the Internet, the original Resource Reservation Protocol (e.g., RSVP [63]) developed in early 1990s, has been extended in order to support security [73], scalability [22], MPLS [74], GMPLS [75] and DiffServ. However, the protocol and its derivatives were not designed for more general signalling services, as they fail to accommodate new signalling needs [20]. Hence, in 2001, IETF formed a new working group, the NSIS, to investigate a more flexible and extensible IP signalling architecture and protocols suite with respect to mobility and QoS interoperability [119]. The NSIS was therefore designed using a two-layer model consisting of a generic signalling transport layer supported by the General Internet Signalling Transport - NTLP/GIST [120], which provides transport services to an upper signalling application layer supported by the NSIS Signalling Layer Protocol - QoS NSLP [121].

Hence, GIST properly indexes and manages control messages transport (e.g., reserve, response, etc.) on behalf of various NSLP signalling protocols, by using a 3-tuple which consists of an NSLP identifier (NSLPID), a Session Identifier (SID) and a Message Routing Information (MRI). In particular, the NSLPID uniquely identifies NSLP protocols since a node is allowed to run several NSLP protocols (e.g., QoS NSLP and NSLP for NAT/Firewall traversal), the MRI describes the flow or the set of flows to which the signalling applies, and the SID indexes the signalling application states in all the NSLPs. This way, the state information is decoupled from the IP address, so changes in IP addresses (e.g., due to mobility, etc.) do not impose complete tear down and re-initiation of a signalling application state (the state parameters, especially the MRI, may be simply updated). Moreover, GIST employs three-way handshake techniques to perform Messaging Associations with adjacent GIST peers, and install security and forwarding state to be used between the peers for each SID. Besides, it is responsible to report nodes/links failure to NSLP, for the later to decide on how to handle failures. While a signalling message makes a complete round trip either on end-to-end basis or within a limited signalling control scope, NSLP-aware nodes may intercept the NSLP messages by means of User Datagram Protocol (UDP) port recognition in routers (as routers are permanently listening on UDP port) or the IP Router Alert Option (RAO) [122]. Then, the message is interpreted and the node decides on how to process the message.

The QoS NSLP provides a generic signalling means able to install, maintain or remove control state on nodes along communication path across heterogeneous QoS enabled transport technologies. To this end, the signalling is decoupled from the QoS resource reservation Model (QoSM) or architecture. This means that a QoS NSLP protocol can signal a network for QoS reservations independently of the specific QoSM, such as IntServ or DiffServ implemented inside the control domain. By definition, a QoSM consists of a QoS architecture and the related QoS provisioning methods, defining the behavior of the Resource Management Function (RMF) [119], as being the responsible for reserving resource for flows. Hence, a QoSM specifies a set of QoS Specifications - Q SPEC parameters [123] as a common language to express QoS requirements and how resources will be managed by the RMF within a domain or between different domains and QoS models. As long as a domain knows how to perform admission control for a given QoS specification object - QSPEC (e.g., mapping data flow to appropriate CoSs), the QoS NSLP does not care about how the specified constraints are enforced and met, since the particular QoS configuration is up to the QoSM of the domain.

As illustrated in Figure 2.5, a media source initiating an NSIS signalling to request a session establishment with specific QoS requirements adds an initiator QSPEC, which indicates the QSPEC parameters that must be interpreted by the downstream nodes unless the reservation fails, thereby ensuring that the intention of the NSIS initiator is preserved along the signalling path. Note that a source or a destination in this example can be a single user or another network upstream or downstream domain. The signalling messages are mainly used to carry or convey the initiator QSPEC, and/or other specific objects such as Record Route Object (RRO) [124] as opaque objects to GIST, which assures the transport so each domain (e.g., QoSM A) intercepts messages and properly processes the relevant objects. As one can see in Figure 2.5, the initiator QSPEC object is translated at the entrance of the QoSM A domain to adapt to the local QoSM specifications, in a way to allow local QoSM-specific RMF to understand and process the required QoS, so as to assure that equivalent QoS is assured for the flow. In order to allow the downstream domains to provide the QoS according to the original QSPEC, the original QSPEC must be kept as is and forwarded to downstream domains which will process the message accordingly.

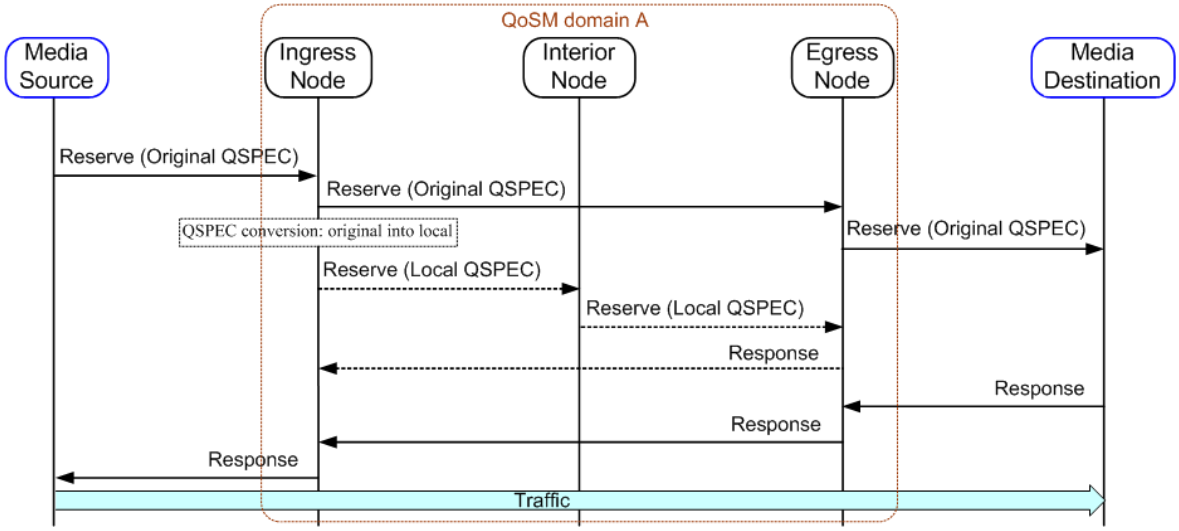


Figure 2.5. Illustration of sender initiated reservation signalling.

In particular, the QSPEC defines four main objects: (1) QoS desired which specifies QoS required by the session requestor; (2) minimum QoS which indicates the minimum QoS to be assured for the services; (3) QoS Available provides the maximum available QoS capabilities of a path; (4) QoS reserved which describes the reservation committed for the requested service. In order to obtain the QoS Available as the resources currently available on a path, each visited node on the path inspects all parameters of the QoS Available object, carried in the signalling message, and if resource available on a node is less than what a particular parameter indicates from the previous nodes, the node updates the parameter in the QSPEC object accordingly. Hence, at the last recipient of a message, the QoS Available object reflects the resources currently available on the bottleneck of the path [125], [30], [126]. Moreover, NSIS supports sender- or receiver- initiated reservations, and bi-directional reservations, and reservations between arbitrary nodes (e.g., edge-to-edge, end-to-access, etc).

Another important object in a QoS NSLP message is the RRO used to build a sequential list of uniquely defined IDs of all nodes on a path. This list may base on, but not limited to, the IDs of the nodes, outgoing interfaces of the nodes on the path, and/or label of the nodes on the path in MPLS-based networks. The RRO is usually built as in the following. The initial RRO contains only one sub-object - the sender's ID. In MPLS-enabled networks, while a node can also collect the switching labels along a path, the labels should not be recorded without the related nodes' ID (e.g., IP addresses) as further details can be found in [74]. In the literature, the concept of RRO object is used for many network optimization purposes. While it serves for explicit route (e.g., specification of groups of nodes or group of ASs to be traversed from a source to a destination) and allows for detecting route changes (e.g., when next hop indicated by RRO differs from that in the Routing Information Base - RIB), it also allows for detecting routing loops. In MPLS-based networks, a

PLR's (Point of Local Repair) such as an ingress node maintains the RRO information as being all the interfaces attached to the tail-end of the backup tunnel. Hence, a PLR exploits this knowledge and the topology database to find the merging point and suitable backup tunnels, by simply comparing the node-ids present in the RROs of both the protected and backup tunnels to improve control performance. Many examples of efforts to extend NSIS exist to support multicasting [30], Inter-Domain Reservation Aggregation in support for large-scale deployment of the QoS NSLP [127] and path-decoupled signalling and QoS reservations [128].

It turns out that NSIS, among other protocols, provides a generic IP signalling platform with more flexible and extensible architecture, which is very important to address QoS and network interoperability in the NGN. Therefore, we exploited this potential and developed an NSIS compliant signalling protocol to support the mechanisms proposed in this Thesis.

2.5 *Next Generation Networks Overview*

As the number of applications to support and their requirements increase, it becomes quite inefficient to provide specialized mechanisms for session control, connectivity control, middleware, signalling, as any single network is usually optimized for some particular services only. These limitations of the traditional design to cope with innovative and enhanced services and applications motivated the research community towards the NGNs [129]. Commonly built around the Internet Protocol, the NGN is a packet-based network able to provide broad range of services including Telecommunication Services. It has been approved that a multitude of different network topologies will have to co-exist or be inter-connected in the future, to optimize the overall network performances and services in a heterogeneous networks environment [130]. The aim is to provide the necessary service capabilities to support present and future multimedia applications and services, and enable enhanced development of new and attractive types of services. In this sense, future network intelligence will no longer just relate to the creative routing of connections based on simple database look-ups, but may take on a much broader meaning (e.g., intelligent management/operations of sessions, multi-technology connections), advanced security, true user agents, user-installable scripts/applets, on-line directory services, and proxy agents). Therefore, an NGN shall simultaneously support wired (e.g., Ethernet) and/or wireless technologies (e.g., WiFi, WiMAX, beyond 3G, 4/5 Generation (3G, 4G/5G) networks).

As illustrated in Figure 2.6, there is a more defined separation between the service/session control and the underlying transport (connectivity) elements of the network, shielding users from the complexity of information gathering, processing, customization, and transportation. Thus, whenever providers want to enable a new service, they can do so by defining it directly at the service layer without considering the transport layer details. End users terminals (e.g., multihomed

terminals with multiple access interfaces) will have access to different service providers and technologies with generalized mobility support, which will allow consistent and ubiquitous provision of services to users. Besides, functions of gateway, registration, authentication and authorization at service level are performed through application support functions and service support functions (ASF&SSF), which work in conjunction with the Service Control Functions (SCF) to provide end users and applications with the services they request. Further, the SCF uses a functional database to accommodate service user profiles as a combination of the user information and other control data into a single user profile function. Then, the Content Delivery Functions (CDFs), under the SCF are responsible for receiving content from the ASF&SSF, storing, processing and delivering it to the end users based on the transport functions capabilities.

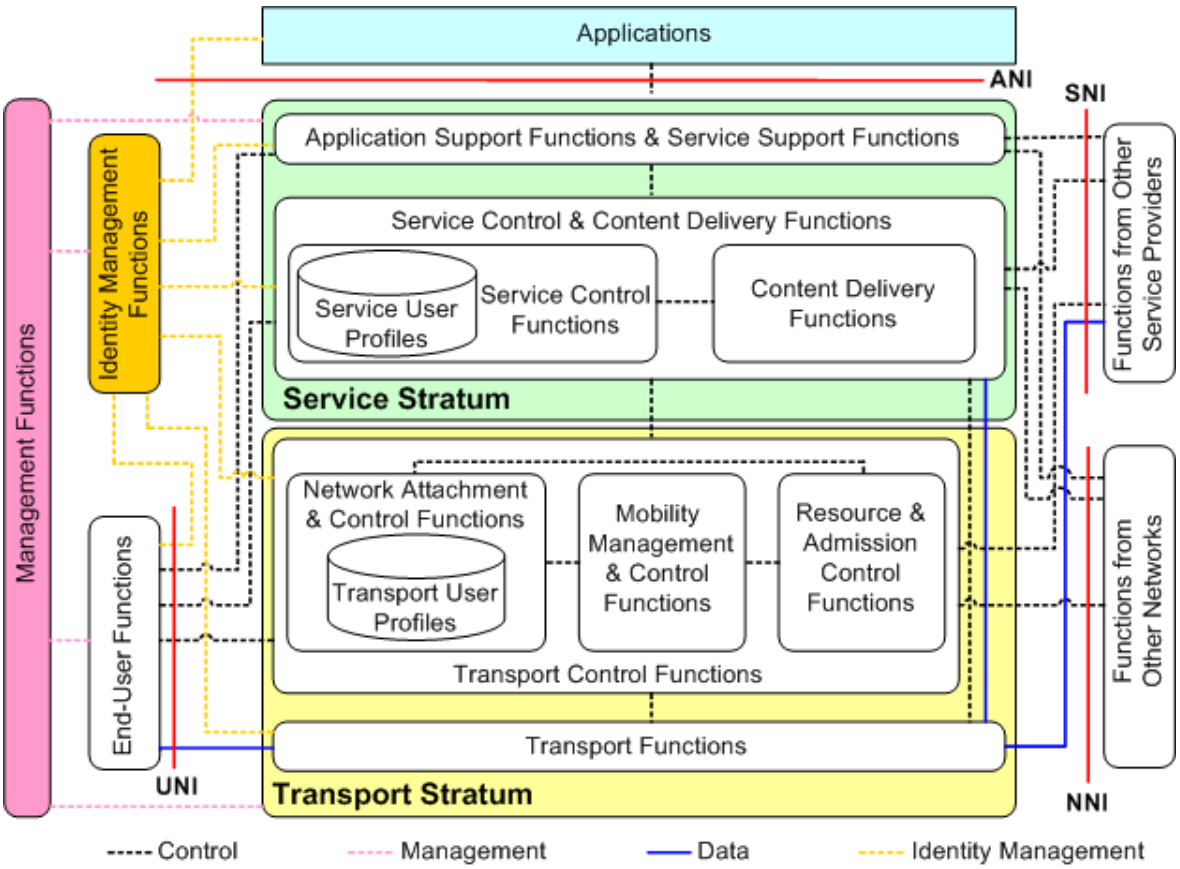


Figure 2.6. NGN Architecture Overview (ITU-T Y.2012).

In the Transport Stratum, the Transport Functions basically provide connectivity for all physically separated functions in terms of the transfer of media, control and management information. In contrast, the Transport Control Functions (TCF) encompass Network Attachment Control Functions (NACF), which hosts user profile at transport layer as a functional database, combining a user's information and other control data into a single "user profile" function (e.g., transport layer level identification/authentication of IP addresses, registration and initialization of end-user functions for accessing NGN services). It also includes Mobility Management and Control

Functions (MMCF) [131] to support mobility within and between its various access network types, and mobility technologies as details on the mobility management requirements are available in [132]. Moreover, the TCF include RACF which bridges between the SCF and the Transport Functions, and provides the SCF with an abstract view of the transport facilities such as network topology, connectivity, resource utilization and QoS mechanisms/technology, etc. Upon the request of the SCF, RACF determines the transport resource availability and admission, and instructs the transport functions to enforce the policy decision, including resource reservation, admission control and gate control, firewall control, etc. It is also responsible for controlling the following functions such as packet filtering; traffic classification, marking, policing, and priority handling; network address and port translation taking into account the transport networks capabilities and the subscribers transport subscription information. Furthermore, the Management Functions (MF) provide the ability to manage the networks for service provisioning with the expected quality, security and reliability, while the Identity Management (IdM) functions assure the identity of entities and support business and security applications (e.g., access control and authorization) including identity-based services.

Hence, the separation of service control from transport functions in the NGN has led to the introduction of Resource and Admission Control (RAC) between the service control and the bearer transport layers to assure QoS. Thus, RAC is responsible for hiding the details of transport network to the service layer, and detecting the resource status of the former to ensure proper and reasonable usage of the transport network resources. Therefore, RAC is a key component that must be well investigated and designed in a way to provide acceptable QoS level to applications by guaranteeing sufficient available resources without wastage or undue control overhead.

2.5.1 Resource and Admission Control Standards

While session demands in a network are generally unpredictable, they mostly request predictable QoS through the network. This situation strongly imposes that network resource and session admission control functions must be carefully performed, since the demands can occasionally exceed the capacity offered by the network. In this sense, the primarily standardized QoS control architectures in the scope of the NGN include the Resource and Admission Control Sub-System (RACS) [133] architecture of the European Telecommunications Standards Institute/Telecommunications and Internet converged Services and Protocols for Advanced Networking (ETSI/TISPAN), the RACF (ITU-T) [134] and the IP Multimedia Sub-system - IMS (3GPP) [135], as further detailed in the following.

2.5.2 **RACS QoS and Admission Control Architecture**

The main objective is to introduce the QoS and Admission Control reference architecture of the ETSI ES 282 003 [133] ETSI ES 282 001 [136] ETSI TS 183 060 [137], known as TISPAN; NGN Functional Architecture; RACS.

2.5.2.1 Architecture Overview

As shown in Figure 2.7 (extracted from ES 282 003 [133]), TISPAN QoS control architecture is composed by four main sub-systems: (1) Application Function (AF) sub-system is the functional entity that shall provide explicit session description with session QoS requirements to express the service expected from a network; (2) RACS is the TISPAN NGN sub-system responsible for the implementation of procedures and mechanisms to handle policy-based resource reservation and admission control for both unicast and multicast traffic in access networks, core networks and customer premise networks, enabling applications to request and reserve resources from the transport networks within the scope of RACS; (3) Transport Processing Functions sub-system that includes basic elementary functions supporting resource reservation enforcement, packet forwarding and routing, and more specific group of functions defined as functional entities; (4) The Network Attachment Sub-system (NASS), which is used for dynamic provision of IP addresses, other terminal configuration parameters as well as authorization of network access based on user profiles.

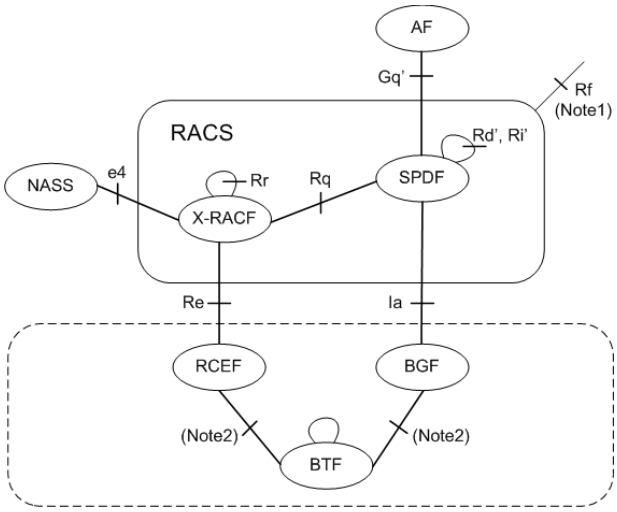


Figure 2.7. TISPAN RACS reference architecture.

The following subsections provide an overview of the main functional entities and interfaces that compose each of the sub-systems shown in Figure 2.7.

2.5.2.2 Application Function

The AF is a sub-system that maps application layer QoS information into appropriate QoS request information to be sent to the Service Policy Definition Function (SPDF) on the Gq' reference point [133] in order to request a service. In particular, it provides explicit session description and session QoS requirements (e.g., IP realm identifier, requestor name/service class, media description, and service priority) in a way that expresses the service expected from RACS. Thus, the requests may comprise Resource Reservation for new session, Resource reservation modification (e.g., quality upgrading or downgrading) for ongoing sessions, and Resource Release for terminated sessions.

2.5.2.3 Service Policy Definition Function

The SPDF is a functional element responsible for making policy decisions (e.g., service authorization) on service requests to the network, based on the local service policy rules defined by the network operator. It carries out a coordination function between the AF through Gq' reference point, the generic RACF (x-RACF) through Rq reference point, the Border Gateway Function (BGF) through *Ia* reference point, the interconnected SPDFs through Rd' in case of intra-domain SPDF or through Ri' for inter-domain SPDF, or any combination of them. The SPDF is also responsible for providing charging information for the Request/Modify/Release/Abort commands via the Rf reference point, upon need. It may reside either in the access as well as in the core administrative domains. In case the authorization of a request is successful, the SPDF sends the requested service along with the service requirements received (e.g., from AF) to the x-RACF, to the BGF, to the interconnected SPDF, or any combination of those, according to the local control policies.

2.5.2.4 Border Gateway Function

The BGF is a control element which provides the interface between two IP-transport domains for user plane media traffic and resides at network boundary (e.g., a gateway). It operates on micro-flows and may implement traffic conditioning functions (e.g., QoS marking, traffic shaping, bandwidth limiting, bandwidth usage metering, etc.) as well as address latching and NAT. In addition, it handles a pool of IP addresses/ports, provides an address independent media session identifier, the address information may change during the media session, and also acts as a dynamic gate to open/close for particular flow according to the instructions received from the SPDF.

2.5.2.5 x-Resource and Admission Control Function

The x-RACF entity receives requests for resources (e.g., bandwidth) from the SPDF via the *Rq* reference point in the Push mode, or from the Resource Control Enforcement Function (RCEF) via the *Re* reference point in the Pull mode, indicating the desired QoS characteristics (e.g., bandwidth,

IP Realm Identifier, Requestor Name/Service Class, etc.). The Push mode consists of the scenario where the RACS "pushes" traffic policies to the transport functions to enforce its policy Decisions. In Pull mode, RACS may receive request from the transport processing functions and then provide traffic policies to the transport processing functions. Hence, the request from the transport processing functions may itself, for example, be triggered by path-coupled requests coming from user equipment and/or transport network elements. Further details on Push or Pull modes are available in ES 282 003 [133]. Then, RACF performs Admission Control based on information such as session QoS information, user profiles received from NASS via *e4* interface, and network resource availability obtained based on its view on the underlying network topology and the related resource status. In this sense, RACF may have a complete or partial view of the network topology and the related resources, including congestion point(s) and the current reservations, etc. Thus, in order to reserve resources, readjust resources, or release resources on the Transport bearer upon need, the so-called *dynamic QoS reservation control*, RACF derives the resource reservation policies (e.g., bandwidth reservation or over-reservation processing functions) and sends appropriate instructions (e.g., appropriate command or signalling) to the RCEF, which is responsible for enforcing policies (e.g., reservation installation, maintenance, readjustment, removal, etc.) on transport infrastructures. Thus, proper QoS control decisions are enforced along communications paths across a network to assure QoS-aware unicast or multicast service delivery over session lifetime. The RACS is also enabled to send the Charging Information directly to the Charging Functions through the Rf reference point, except the offline charging which may terminate on both x-RACF and SPDF (e.g., pull mode).

It is important to note that a transport segment may have multiple instances of x-RACFs, and that each x-RACF may be involved in resource admission control for unicast services, multicast services, or both (each of them may have a complete or partial view of the network topology and/or resources). Hence, the x-RACF involved in controlling the same transport resource (e.g., a path), shall be arranged in a tree structure (top tier and lower tier) where the top tier x-RACF in this structure is the one interacting with SPDF. Moreover, the Rr reference point allows x-RACF instances to cooperate/synchronize with each other on the topology, the allocated and available resources to avoid uncontrolled overbooking while reserving resources spanning multiple transport segments. Such design needs to be carefully addressed since synchronization between x-RACF instances can be achieved at various granularity (per-flow or not), thus allowing for different trade-offs between synchronization overhead and the resource sharing efficiency.

2.5.2.6 Resource Control Enforcement Function

The RCEF is a transport processing functional entity that performs L2/L3 policy enforcement functions for unicast and/or multicast under the control of the RACF through the *Re* reference point. Hence, RCEF either enforces the policy autonomously or in conjunction with the Basic Transport Function (BTF) to trigger service transport control actions. In this sense, RCEF may interact with BTF to enforce policies which impact data forwarding behaviour, such as data replication for multicast traffic. Besides, RCEF includes functionalities such as resource allocation for upstream and downstream traffic. Moreover, depending on local control policies, RCEF may notify RACF about certain events (e.g., feedback messages, links/nodes failures, control malfunction, etc.) occurred on the network elements. Then, RACF may decide to modify/remove existing policies, install new policies or escalate the event to higher layer.

2.5.2.7 Basic Transport Function

The BTF is an integral part of all network transport segments. It implements the so-called Elementary Forwarding Functions (EFFs) which is used for traffic flows forwarding, and the Elementary Control Functions (ECFs) used to process control protocol data for unicast as well as multicast (e.g., control signalling, routing protocol, etc.). Hence, the ECF on a node might decide to send control protocol data to other ECF, or interact with one or more EFF to establish new or modify existing forwarding behaviour by manipulating the related routing or forwarding databases on nodes. In general, almost all physical network elements (e.g., a bridge, a router etc.) typically contain a BTF and might contain additional functional entities such as RCEF.

2.5.2.8 Network Attachment Sub-system

The NASS provides attachment information such as dynamic provision of IP addresses, terminal configuration parameters, the authentication taking place at the IP layer prior or during the address allocation procedure, the authorization of network access based on user profiles, and location management taking place at the IP layer.

2.5.3 RACF QoS and Admission Control Architecture

The generic architecture of RACF [134] specified by the ITU-T in Y.2111 is illustrated in Figure 2.8. In particular, two primary Functional Entities (FEs), being the Policy Decision Functional Entity (PD-FE) and Transport Resource Control Functional Entity (TRC-FE), have been defined. The PD-FE is responsible for making the final session admission decision based on the network policy rules, the session information received from the SCF via *Rs* interface, the transport subscription profile provided by the NACF through *Ru* interface in access networks, and the resource-based admission decision results obtained from the TRC-FE on the *Rt* interface. For this purpose, the TRC-FE collects network topology and related resource status information (e.g., using

COPS or SNMP protocol) via the *Rc* interface and provides resource-based admission control decision results to PD-FE. Besides, the Policy Enforcement Functional Entity (PE-FE) is a gateway which can be located between the Customer Premise Equipment (CPE) and Access Network (AN), AN and Core Network (CN), CN and CN. It receives intructions from the PD-FE via the *Rw* interface and performs the transport functions (e.g., in routers) such as gate control, bandwidth allocation, rate limiting, IP packet marking, Network Address and Port Translation (NAPT) control, etc. Further, the Transport Resource Enforcement Functional Entity (TRE-FE) enforces transport resource policy rules as instructed by TRC-FE through the *Rn* interface.

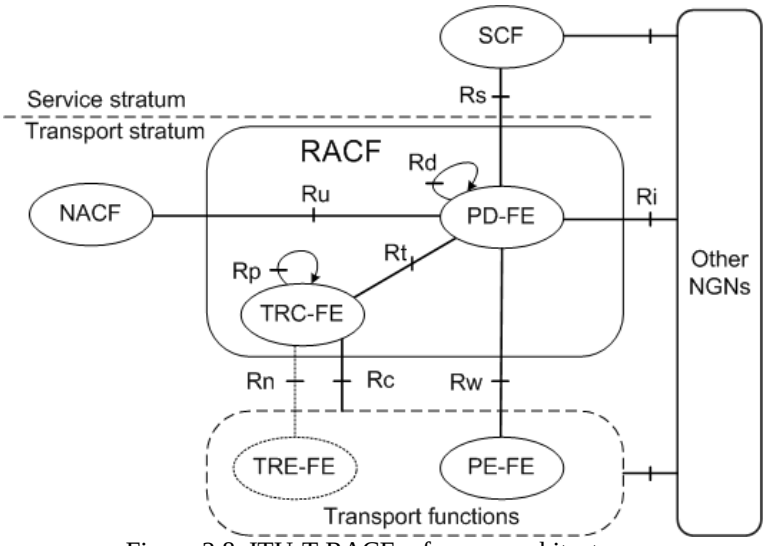


Figure 2.8. ITU-T RACF reference architecture.

Thus, the RACF architecture is consistent with that of the RACS detailed in subsection 2.5.2. This implies that, the PD-FE and the NACF are similar to the SPDF and the NASS respectively, while the TRC-FE and the PE-FE can be considered as in correspondence with the x-RACF and BGF (including the RCEF), respectively, even though there are differences in the functions and reference points.

2.5.4 IMS QoS and Admission Control Architecture

The IMS [135] architecture has been introduced by 3GPP to provide a secure and reliable means for terminals and applications to negotiate and communicate with each other. It is access independent i.e. the same service can be provided over various access technologies, and allows for tracking services to enable flexible means for charging such as the traditional time-based as well as packet and service-based charging. As illustrated in Figure 2.9, the "core" of IMS in the service stratum encompasses two main elements: the Call Session Control Function (CSCF) and the Home Subscriber Server (HSS).

The HSS is a database that maintains user and subscriber information to provide the following functions: identification handling, access authorization, authentication, mobility management

(keeping track of which session control entity is serving the user), session establishment support, service provisioning support. Besides, the CSCF is mainly used to assure session control for terminals, and applications using the IMS network and can play three different roles: Serving-, Interrogating- and Proxy- Call Session Control Function (S-, I- and P-CSCF). The P-CSCF is the first point of contact for users with the IMS system and the I-CSCF is the first point of contact between peered networks. Hence, while the P-CSCF is used to ensure security of the messages between user and the network, or to allocate resources for the sessions, the I-CSCF is responsible for querying the HSS to determine the S-CSCF for a user and may also hide the operator's topology from peer networks. Further, the S-CSCF is key component which allows for processing registrations and maintaining record of users' location, authentication, and the sessions processing based on control policy stored in the HSS.

Besides the IMS application plane in the service stratum which is not included in RACF or RACS, the resource and admission control in the transport stratum using RACF, RACS, or IMS is commonly implemented by two FEs, namely, the PDP and the PEP. This policy-based QoS management provides complete network control, including traffic congestion management, traffic shaping and policing, bandwidth control and traffic balancing [21]. In the transport stratum, these standards differ mainly in QoS coverage, terminology, and the corresponding FEs and the main functions usually deployed for the resource and admission control in the transport stratum are detailed using the RACS architecture in subsection 2.5.2. It is important to mention that the aim of the RAC functional architectures is to provide a general resource control framework which is independent of the physical implementation of either access or core networks.

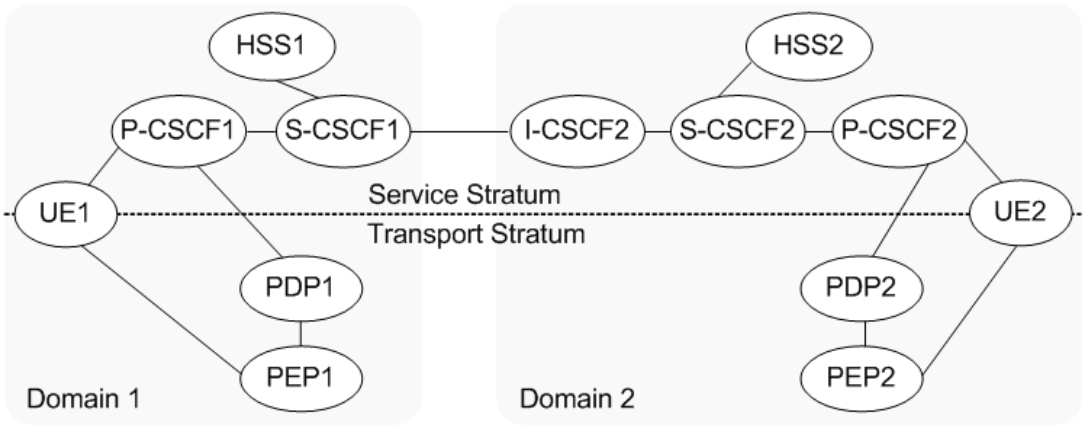


Figure 2.9. 3GPP IMS architecture overview.

In order to facilitate the understanding of possible interactions between the service stratum and the transport stratum in the NGN, we use Figure 2.9 to illustrate a successful session setup between two users located in different administrative domains. Hence, suppose that a User Equipment in a network domain 1 (UE1) (please Figure 2.9) wants to establish a QoS-sensitive session connection

with a UE2 in a network domain 2. First, UE1 specifies its QoS parameters in a SDP and sends a SIP message (i.e., INVITE) to the P-CSCF1. After authentication and security checks, P-CSCF1 forwards the SIP message to the S-CSCF1 for authorization of the session requested based on the service policy and the registration status of the UE1 stored in the HSS1. Afterwards, the message is forwarded to the I-CSCF2 at the entry point of domain 2, which in turn sends it to the UE2 through the P-CSCF2. Then, UE2 also defines its desired QoS parameters and sends a SIP response to the UE1 via the same IMS signalling path. Thus, UE1 and UE2 repeat this SIP message exchange until they agree on a set of QoS parameters to be used for the communication. Then, the P-CSCF1 consults the PDP, triggering the resource and admission control process in the transport stratum. To this end, the PDP decides whether to grant or deny the request by taking into account the requested QoS parameters, the UE1 profile, network current resource availability, and local control policy. Upon successful operations, the PDP1 maps the negotiated QoS parameters to its local QoS parameters semantics, and instructs the PEP1 to enforce the reservation when the user requests it. In this case, P-CSCF1 forwards the SIP message to UE1, informing that the latter can request resource reservation in the transport stratum. As UE1 initiates reservation request to the PEP1, it sends a SIP message to UE2 so that the latter can start requesting its resource reservation process after replying to the SIP message. In terms of comparison of IMS RAC functions with those of RACS and RACF, the hereinabove illustration shows that, the P-CSCF is enabled to request resources and admission similarly to AF and SCF respectively in RACS and in RACF. Moreover, the PDP is queried to carry out the roles of SPDF and x-RACF in RACS, and those of PD-FE and TRC-FE in RACF architectures. Therefore, the IMS core is compatible with the RAC operations of both the RACS and the RACF.

This study shows that the control of QoS-sensitive session (e.g., interactive video streaming) establishment is commonly divided into two phases: a QoS parameters negotiation, and a network resource and admission control phases. The QoS negotiation is executed in the service stratum where the session participants (e.g., caller and callee) exchange their expected QoS parameters (e.g., bandwidth, type of media, transport protocol, type of codes, delay, jitter and loss requirements). At this stage, the participants must first agree on a set of negotiated QoS parameters before the session can be set up. During the resource and admission phase, which is performed in the transport stratum through the RAC, the network resources required by the negotiated QoS parameters may be granted or denied depending on the QoS parameters, the users' profiles, the current network resource availability, and local control policy [21].

In order to demonstrate the pragmatism of the approaches proposed in this Thesis with the existing standards, we use the RACS reference architecture to implement the ACA functions and

operations which are detailed later in Chapter 3. This intends to facilitate further understanding of how our designs can integrate with standardized architectures.

2.6 *Network Control Models*

This section surveys network control models mainly in terms of centralization and decentralization of networking control.

2.6.1 Centralized Models

The Figure 2.10 illustrates a centralized architecture in which the Central Controller (CC) is in charge of the overall control of the network. Generally, the CC maintains the network topology and is responsible for defining the policies and amount of resources to enforce in the underlying network upon need. Therefore, session demands for network resources should be addressed to the CC, usually through the access points (e.g., gateway, border nodes, etc.).

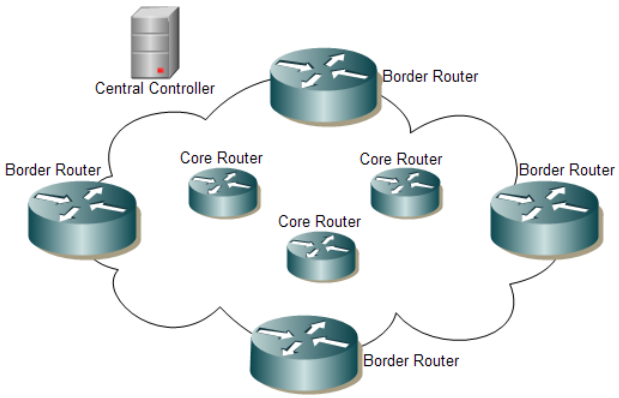


Figure 2.10. Illustration of centralized network scenario.

Many centralized network proposals for resource management, signalling and admission control are available in the literature [138], [139]. The central entities may be referred under different names such as Bandwidth Brokers [140], [141], QoS Brokers [89]. The work in [142], [47], [143], [48] addresses comprehensive end-to-end QoS-aware content distribution chain spanning heterogeneous centralized networks. As each domain deploys its own centralized architecture, the end-to-end control is coordinated hierarchically through SLAs/SLSs between the different domains on the chain. Each network implements aggregate resource control, and the end-to-end provisioning is performed periodically. However, studies [23] show that admission control should not be planned (e.g., based on SLAs and provider SLSs - pSLSs); it needs to dynamically reconfigure parameters taking QoS demands and current network conditions into account to effectively cope with network under utilization. Therefore, the periodic provisioning strategies in these approaches confront serious limitation in terms of waste of resources. A QoS broker control approach is proposed in [144] to demonstrate how the end-to-end QoS control architecture developed by the ITU-T NGN/GSI for the

NGN can be applied to smart grid to assure stringent QoS provisioning in a centralized and standardized manner. One may see this as similar to design in wireless systems due to the unpredictable characteristics of broadband power line (e.g., unpredictable frequency, impulse and background noise and their wide variability, attenuation, limited bandwidth, variability of bandwidth, etc).

In [145], research efforts show that self-organization with policy-based configuration and reconfiguration of IMS components and corresponding nodes, to dynamically adapt themselves based on the features like network load, number of users and available system resources, is necessary to succeed the IMS deployment at reduced cost and complexity. With respect to centralization, a master node maintains operator policy and state information of all nodes under its control, and assigns functionalities and roles to other nodes based on the capabilities of the nodes to dynamically improve performance (e.g., load balance). The master node determines the functional behavior of all other nodes in a network through a periodic messaging mechanism. However, this study is focused on IMS functional components merging, splitting and relocation between IMS-capable nodes without external intervention, and therefore, it is not pragmatic for general network control purpose.

The existing centralized approaches, besides ease to manage, present a single point of failure while the central entity is getting more and more bottlenecked with the explosive growth of network demands and their dynamism. As it is studied in [49], centralized networks are preferable in small scenarios or when the majority of demands are initiated or destined to one edge node. Otherwise, decentralization better fits in large scale network or when traffic load is high and uniformly distributed. This means that the network design must take several input parameters (e.g., network size, number of customers, dynamicity of the QoS request, performance of the control servers, etc.) in its choice for centralization or decentralization.

2.6.2 **Decentralized Models**

Network decentralization paradigm allows for taking control decisions at distributed entities (decision points) throughout a network with no central controller, as illustrated in Figure 2.11. As in [43], distributed control efforts focus on mechanisms for enabling networks with self-awareness, self-optimization, and self management capabilities, whereby network elements can adapt themselves to contextual changes without any external intervention. This way, the control load is distributed across the network, as being a key requirement for scalability. It is also investigated in [13] that a decision point requires a good knowledge of its underlying network topology along with the available resources [11], and their location on communication paths [12] to allow for improving performance in the NGN. Hence, in decentralization environment, synchronization between the

distributed decision points is essential for the latter to maintain consistent control information to avoid wrong decisions, which is very challenging since excessive signalling and the related processing overhead would jeopardize scalability. This has its root in the design philosophy of the Internet Protocols. In [146], Clark stated: *"because of the distributed nature of the replication, algorithms to ensure robust replication are themselves difficult to build, and few networks with distributed state information provide any sort of protection against failure."*.

In the field of multi-agent systems [147], self-organization studies mostly emerge from naturally inspired approaches such as ants, termites and honey bees swarming or immune systems. The bio-inspired self-organizing systems based on reactive agents have been used to implement diverse applications such as ubiquitous service oriented networks [148], distributed coordination of robot and synchronization of their movement to achieve group locomotion [149], emergent forecasting in manufacturing coordination and control systems [150], load balancing [151] and security [152]. Kvalbein et al in [153] propose that network nodes perform forwarding decisions and traffic load-balance over multiple paths, independently, based on their local knowledge of the network control information, so there is no synchronization orchestrated among the distributed nodes. This way, the nodes lack global view of network conditions required to improve performance, and the proposal focuses on robustness and simplicity rather than optimality.

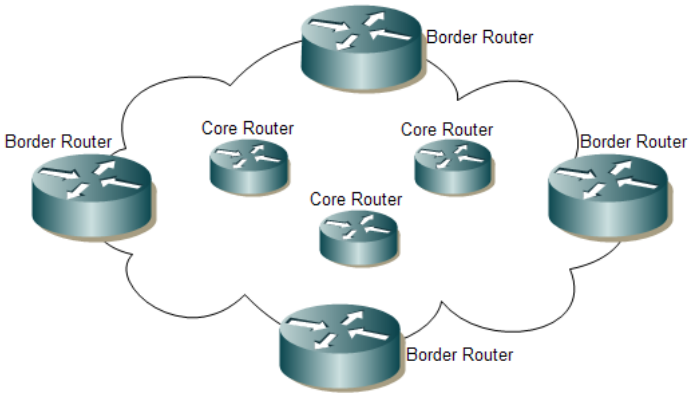


Figure 2.11. Illustration of decentralized network scenario.

The authors in [154] argue on decentralization, self-organization, embedding of functionalities and autonomy as the guiding principles for In-Network Management (INM) to achieve scalable, robust management systems with low complexity for large-scale, dynamic network environments. They propose ways for monitoring network-wide aggregates (e.g., total number of flows, the maximum link utilization, packet drop rate, etc.) in real-time through computation across nodes in a neighborhood, a network domain or the entire network using aggregation functions. In literature, spanning tree based (incrementally computation from leaves to the root) [155], [156] and gossiping based (computation result is available on all nodes and converges probabilistically to the true value) [157], [158] protocols are two main approaches for computing the aggregates in a

distributed way. While continuous monitoring is a major requirement to provide accurate inputs to decision-making processes in these solutions, the signalling and processing overhead increases with the decrease of adaptation time, which poses scalability issue.

A mechanism for resource control among virtual networks using a managed self-organizing network concept, and a dynamic resource allocation technique taking into account the correlation of traffic demand and route diversity has been investigated in [159]. The impact of correlation of traffic of the same source and destination pairs is considered to improve resource utilization for competing virtual networks. This approach focuses on the traffic of the same source-destination pair, but further research is still necessary to address a more generic environment with dynamic correlation of traffic from multiple sources to multiple destinations. Moreover, resource over-reservation could be studied in such systems to reduce control signalling frequency to further improve performance.

A decentralized approach for IMS components and corresponding nodes is introduced in [145] with policy-based dynamic configuration and reconfiguration based on system demands and the network resources, aiming to reduce both deployment cost and complexity. In particular, each new IMS node announces its presence through a multicast message, and existing nodes reply if they need to transfer some of their current functionalities, and the new node may accept transfer request in a first in first out (FIFO) fashion. In case the new node receives no role transfer request, it takes over all roles. A major limitation of this approach is its focus on dynamic IMS functional components merging, splitting and relocation between IMS-capable nodes rather than on a broader application.

In [43], a Decentralized Adaptive Coordinated Resource Management (DACoRM) approach is introduced for intra-domain IP networks resource management. DACoRM enables source nodes, formed by ingress routers in full-mesh or ring topology, to cooperate and exchange information about new configurations inside a network with support for multiple paths between any source-destination pair using the multi topology routing (MTR) protocol [160]. In this proposal, the intra-domain links are statically and logically partitioned into a number of disjoint subsets, which are distributed among the different source nodes, such that each subset is placed under the responsibility of only one source (ingress) for future control decisions making. Every source node achieves this by adjusting splitting ratios of traffic flows computed through a sequence of re-configuration processes, according to link utilization information disseminated through traffic engineering (TE) enabled OSPF which incorporates TE metrics into link state advertisement [81]. Hence, upon receiving information about a highly loaded link, a source node checks whether the link belongs to its subset, so as to assume the role of the new reconfiguration. A node can also delegate a re-configuration task to other nodes in case it is not able to determine the configuration

by itself in such a way that only one source node is permitted to perform a splitting ratio adjustment at a time. Besides complexity, the focus of DACoRM is on load-balance and not a means for overall intra-domain network control. Another drawback here is the periodic cooperation among sources (ingress nodes), since periodic techniques mostly confront a trade-off between signalling overhead, information accuracy and network under-utilization issues.

Wakamiya et al., [44] argued that centralized mechanism is ineffective for considerable maintenance overhead to collect and keep up-to-date and consistent information on a whole network system due to the increase of the number of nodes and the size of network. On the other hand, a self-organizing system may sacrifice performance to some extent to achieve scalability, adaptability, and robustness when each node only communicates with neighbor nodes to obtain local information and not a global knowledge of network status. Besides, self-organization system may take time to converge to become stable, and it would be difficult to maintain and control the whole system. Even though one can make all nodes report their status to a central controller, this would waste bandwidth and energy. Bearing these challenges in mind, "Clean slate" approaches and design requirements for the new generation network can be found in [52]. Therefore, significant research efforts were still expected in the field of the overall network control designs.

2.7 *Scalable Resources and Admission Control Proposals*

The severe issues over per-flow resource reservation approaches [63] has motivated the research community towards aggregate bandwidth resource and QoS control for many years, intending to minimize control cost in terms of signalling, states and related processing overhead. This subsection introduces the most important and relevant proposals within the scope of this Thesis.

2.7.1 IntServ over DiffServ

The Figure 2.12 shows an example of IntServ over DiffServ [161] scenario consisting of a Diffserv region in the middle of a larger IntServ end-to-end enabled network, meaning that the regions outside the Diffserv region contain at least some nodes which support the Integrated Services architecture.

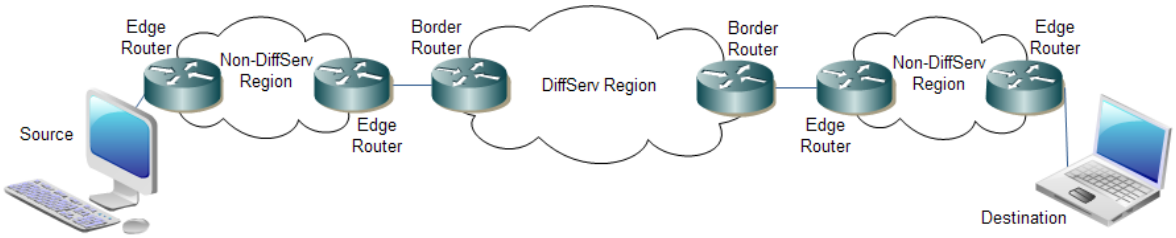


Figure 2.12. Illustration of IntServ over DiffServ network scenario.

In such network environment, end-hosts must be enabled to request per-flow, quantifiable resources, along end-to-end data paths and to obtain feedback regarding admissibility of these requests, resorting to RSVP, before sessions can be established to provide expected QoS. In the IntServ-aware networks outside the DiffServ domain, each individual flow is subject to explicit control functions (e.g., admission control, classification, resource reservation, etc.). In the DiffServ domain, the requests for IntServ services must be subject to classification to select appropriate PHB, admission control that takes into account the availability of resource in the DiffServ domain and admitted traffic must be conditioned (e.g., metering, marking, shaping, policing). Besides, the resource provisioning may be performed in many different ways.

- Resource may be statically provisioned to the existing aggregate traffic according to SLA.
- Resource may be dynamically provisioned to aggregate traffic, resorting to RSVP or by other means such as Bandwidth Brokers.

In the first case, the RSVP signalling is transparently carried across the DiffServ region as the nodes inside the domain may not be RSVP aware. To achieve expected service levels, customer(s) of the Diffserv network regions and the owner of the Diffserv network region negotiate static contract through the SLS, which determines the transmit capacity to be provided to the customer across the Diffserv domain, and exchange of resource availability between two adjacent domains. As indicated in [161], the “transmit capacity” may be simply an amount of bandwidth or it could be a more complex “profile” involving a number of factors such as burst size, peak rate, time of day etc. Although this is scalable, it is quite inefficient and inflexible, and therefore it is not suitable for highly dynamic scenarios.

As an alternative in the second case, the nodes inside the DiffServ domain are able to participate in some form of RSVP signalling and thus, the resource is provisioned dynamically (e.g., increase or reduced) for every PHB depending on the demand of the latter while classifications and scheduling functions are handled aggregately, based on DSCP, not on the per-flow basis. Hence, this approach is more scalable than the pure RSPV/IntServ [1]. However, the signalling per-flow remains a major limitation for scalability and more scalable approaches such as aggregate RSVP was still deemed to be investigated.

There are other approaches which use aggregate resource reservation control, seeking to create scalable networking environment for provisioning and maintaining end-to-end agreed QoS through heterogeneous networks, owned by different operators [142], [47].

2.7.2 **DAIDALOS**

In the context of DAIDALOS QoS approach [89], an administrative domain may contain several access networks (ANs) attached to a core sub-domain via sub-domain routers while different administrative domains inter-connect via Edge Routers (ERs). In this approach, each AN deploys a QoS Broker which performs resource management using per-flow reservation control while the QoS Broker in the core sub-domain manages resource on aggregate basis following DiffServ model. Hence, to achieve fine-grained end-to-end resource and admission control while keeping scalability features in the core sub-domain, QoS information on the aggregates inside the core sub-domain and the inter-domain segments is provided to QoS Broker of core sub-domain through active and passive probing, and then propagated to the QoS Brokers in the ANs.

2.7.3 **ENTHRONE**

Within the European ENTHRONE project [47], a comprehensive end-to-end audio-visual distribution chain, ranging from content generation and protection, and with distribution across QoS-enabled heterogeneous networks to the delivery of content at user terminals have been investigated. ENTHRONE architecture defines a *Service Plane* for appropriate SLAs/SLSs establishment among operators, providers and customers. Besides, a *Management Plane (MPl)* is dedicated for long term actions over resource and traffic management for QoS and control efficiency purposes, while a *Control Plane (CPl)* is used to assure short term actions on resource and traffic engineering control, including routing. In terms of QoS control scalability within Enthrone, an aggregate resource provisioning solution is adopted as available in [139].

In this approach, Service Provider establishes provider SLA (pSLA) contract with one or several content providers, which give the former with information such as the location of content servers, details about the contents and access rights to Digital Items. This allows the Service Provider to negotiate and conclude pSLSs with Network Providers to establish aggregated QoS enabled pipes (traffic trunk) through inter-domain from Contents Servers region towards regions where potential Content Consumers are located (based on estimation on resource usage by customers). Thus, a virtual network provisioned with aggregated pipes (e.g., bandwidth and PHBs) is established at inter-domain level in advance to avoid per-flow QoS reservation signalling. Such aggregate resources provisioning is negotiated and redimensioned periodically as defined by the network management. While large periodic for aggregate pipes re-dimensioning is necessary to optimize signalling frequency, it would lead to significant waste of resources especially when deploying multiple edge nodes. On the other hand, short periodic operations would place undue signalling overhead. Kashihara et al. [23] studied that network resource and admission control should not be planned (e.g., based on SLAs/SLSs) to achieve optimization of resource utilization, it

should be dynamically reconfigured by taking current need of resource usage into account. Moreover, Enthrone is a centralized approach which raises other limitations such bottlenecked central station in large scenarios or single point of failure issue.

2.7.4 **EuQoS**

The IST European Project EuQoS [48] also sets its objective to build, integrate and validate end-to-end scalable QoS provisioning across different administrative domains, and heterogeneous networking technologies for advanced QoS-aware multimedia applications. EuQoS system emphasizes on preserving the Internet openness by providing a simple multi-protocol interface implemented with Simple Object Access Protocol - SOAP that allows end users to request any kind of service without being required any specific application signalling protocol (e.g., SIP, H.323, etc.), in contrast to IMS, which requires the usage of SIP protocol to interact with the P-CSCF (Proxy Call Session Control Function) [143]. The architecture is divided into Service, Control and Transport Planes along the vertical axis, and the horizontal axis encompasses various technologies spanning Core and Access Networks. Hence, the service plane enables customers to request session establishement/release/modification, while a Diameter [112] server is in charge of Authentication (managing the user access to network resource), Authorization (granting services and QoS level to the requesting user) and Accounting (collecting accounting data) with support for roaming features. Based upon centralized approach, the control plane manages and provides end-to-end QoS paths (EQ-Paths) assured through inter-domain coordination, and maintains network topology and monitors the resources usage information by means of traditional measurement tools. In particular, intra-domain control and policy decisions are taken by a central entity, and the decisions enforcement and devices configurations inside the domain are performed using COPS protocol while inter-domain QoS requests process is based on NSIS protocol.

From scalability perspective, EuQoS system bases on the concept of end-to-end QoS-link (EQ-link), consisting of virtualizing border router to border router link which explicitly establishes known QoS characteristics (e.g., bandwidth, buffer) to specific CoS (not to any specific session) in order to carry aggregate traffic. Then, the end-to-end paths (EQ-Paths) provisioning is carried out periodically (in order of hours or days) by selecting appropriate EQ-links between various networks. The best path computation is dedicated to a central PCE (Path Computation Element) server which uses hierarchical approach to scale by managing only a fixed amount of tunnels (e.g., no more than a full mesh of tunnel per CoS), independently of the number of ASs on the end-to-end paths. Hence, upon receiving a service request from a user, the admission process selects the appropriate EQ-Paths aiming to minimize congestion occurrence. Major limitations of this solution include not only the periodic EQ-Paths provisioning which confronts wastage resources issues, but

also the use of traditional measurement tool while being centralized. As we described in sub-section 2.2, path probing places undue signalling overhead which needs careful attention.

2.7.5 **Q3M**

The recognition of the limitations of centralized solutions to provide QoS-enabled access and connectivity to network supporting seamless mobility for multi-user sessions across heterogeneous wired and wireless environments motivated the advent of the *QoS architecture for Multi-user Mobile Multimedia* (Q3M) architecture [162]. Q3M system pushes control intelligence to network edge nodes (network border nodes) configured as network CDPs, and core nodes are used mainly for the decision enforcement, using a modular approach without a central controller. The Q3M architecture integrates three main components denotes as Cache-based Seamless Mobility (CASM) [163], Multi-service Resource Allocation (MIRA) [30] and Multi-user Session Control (MUSC) [164]. While CASM provides seamless mobility to users between heterogeneous clusters, MIRA controls intra-domain network bandwidth and multicast trees resources for multi-user session distribution based on the DiffServ model, and the MUSC performs QoS mapping, QoS adaptation and connectivity control for fixed and mobile users ubiquitous access in heterogeneous network environments. In particular, every edge node implements the MUSC and MIRA components and a core node only hosts MIRA. In addition, the edge nodes located at access-domain include the CASM to deal with mobility. MIRA allocates surplus of multicast distribution trees in advance and controls connectivity on-demand to improve flow re-routing. However, its per-flow QoS reservation signalling approach is not scalable.

2.7.6 **MARA**

In order to address this issue within Q3M, the *Multi-user Aggregated Resource Allocation* (MARA) [125], [162] was introduced with a set of functions to dynamically control bandwidth over-reservation for CoSs inside each network domain. The MARA resource over-reservation algorithm is embedded in edge nodes where resource over-reservations parameters are defined, and the decisions are conveyed in NSIS compliant signalling message to be enforced on core nodes inside a network domain in which resides the corresponding edge node. The operations of MARA are divided into network initialization phase and running phase. At network initialization phase, MARA defines a peak threshold χ_i and a certain amount of bandwidth reservation B_i for each CoS_i inside a network domain. Then, as long as χ_i is not exhausted and the reservation of a CoS_i is exhausted, it computes new surplus of reservation B_i for the CoS_i. In MARA, an ingress edge node maintains resource utilization statistics of paths' bottleneck interfaces only, which information is acquired using periodic and on-demand probing operations. This implies that MARA also

confronts paths probing problems. Further, when a threshold χ_i is exhausted, it readjusts all the thresholds dynamically by attempting to grant a congested CoS with a portion of residual bandwidth from each of the remaining CoSs. As we studied later in this Thesis, these MARA's resource computation functions, in essence, suffer from waste of bandwidth or unnecessary blocking, especially when network is near congestion, and thus fails to efficiently utilize resources. Moreover, there is no synchronization mechanism between the edge nodes in Q3M architecture while MARA does not provide any information on how multiple distributed edge nodes could effectively maintain consistent resource utilization statistics information about core nodes to prevent control inconsistency which could lead to QoS violations, waste of resources and therefore unnecessary increase of session blocking probability. Kashihara et al. [23] demonstrate that, in a network domain which deploys multiple edge nodes, a central entity is mandatory for a proper resource management at edge nodes without per-flow signalling on core nodes unless the edge nodes are well synchronized to avoid incorrect decisions.

2.7.7 Aggregate Resource and Admission Control Schemes

In terms of scalable resource reservation control, Pan et al. [165] proposed to delay resource release events and to over-reserve bandwidth surplus as a multiple of a fixed integer quantity, namely "quantization", in the Border Gateway Reservation Protocol (BGRP) for aggregate flows destined to a certain domain - a Sink-Tree-Based Aggregation Protocol. This solution does not comply with network dynamics, and therefore, it fails to efficiently utilize the network resources. Further, Sofia et al. [166], [167] proposed the use of bandwidth over-reservation to reduce excessive QoS signalling load of the Shared-segment Inter-domain Control Aggregation Protocol (SICAP). In SICAP, the authors provided valuable analysis of over-reservation schemes in a broad range of settings and compared results with previous representative solutions such as the BGRP. In particular, BGRP and SICAP rely on path probing techniques to acquire resource statistics on bottleneck links and prevent over-reserving too much resources; the more resources they over-reserve, the more session requests are blocked unnecessarily. Moreover, these are inter-domain aggregation control solutions, in contrast to ACOR which is an intra-domain approach.

The Simple Inter-Domain QoS Signalling Protocol (SIDSP) [168] system suggests to over-provision virtual trunks of aggregate flows based on a predictive algorithm (e.g., based on past history) without appropriate mechanism to dynamically control the residual bandwidth between various trunks. Besides being an inter-domain mechanism, SIDSP focuses on aggregation of reservations, not on over-reservation, and demonstrates superiority over BGRP in terms of reservations state load reduction on border routers implemented in transit Autonomous Systems (ASs). Furthermore, the Dynamic Aggregation of Reservations for Internet Services (DARIS)

[169] proposed to aggregate reservations along ASs paths to reduce stored states and studied bandwidth over-reservation as a means to prevent excessive signalling overhead when compared with protocols which do not perform aggregation. However, DARIS is a centralized solution which uses central entities similar to bandwidth brokers and thus presents scalability as well as single point of failure problems.

Prior et al. [170] investigated the inefficiency issues confronted in over-reservation proposals by comparing per-flow reservation solutions with the IETF proposed aggregate reservation [22] approach. They found out that, per-flow signalling allows for preventing the wastage of resources at the expense of undue signalling overhead and thus, fails to scale. Besides, while aggregate approach can reduce the signalling overhead by means of over-reservations, it increases session requests blocking probability unnecessarily with the increase of the over-reservations. Hence, further research was still deemed necessary to address this major trade-off, that is, whether a reasonable operating point can be achieved to allow for over-reserving as much as necessary to significantly reduce signalling overhead without wasting resources.

The work in [171] proposes an overprovisioning-centric and load balance-aided solution called QoS-RRC, using the MARA [125] described earlier in sub-section 2.7.6 and shows interesting results. Although the QoS-RRC deploys a central server called Generic Path Factory, each ingress router hosting a QoS-RRC decision agent, simply performs control in a decentralized manner. Each ingress router deploying the MARA algorithm decides and readjusts over-reservation parameters independently and dynamically on links of which the resource is shared by all the ingresses, while there is no cooperation mechanism between the ingress routers to jointly exploit appropriate information to improve performance. Another over-reservation proposal is available in [172]. Besides being centralized, this work relies on traditional periodic paths probing techniques to acquire resource conditions on network bottleneck links as in C-CAST project [173], and the resource utilization statistics are not recorded on real-time basis for all the interfaces used inside the network. Hence, the inefficiency of over-reservation computation functions of MARA [125] (detailed earlier) together with signalling overhead and inaccuracy issues of probing techniques used in C-CAST project expose the approach to QoS violation and waste of resource problems. It is studied in [24] that, although aggregation of reservation allows for reducing both control signalling and state overhead, it needs to be carefully designed since inefficient solution incurs network under-utilization or waste of resource, while the number of aggregates to be maintained can still be quite large in a network with many edge routers.

Although the aggregation of reservations allows for reducing QoS control signalling and state overhead, the trade-off imposed in terms of waste of resources is a major challenge for aggregate resource control. In distributed scenario, it is generally argued that, increasing the number of

distributed control entities improves scalability and reliability, but at an eventual cost of coordination between distributed entities, the impact of resource fragmentation [24], [25] and waste of resources. Hence, many recent proposals have focused on using per-flow admission control signalling mechanism [174], [89] with reduced states overhead (improved scalability feature) instead of admission based on aggregate resources over-reservation. In this sense, the Resource Management in DiffServ [175] was introduced as a dynamic resource reservation control within DiffServ network domains. In particular, it proposed two main protocols designated Per Domain Reservation (PDR) protocol and Per Hop Reservation (PHR) protocol.

The PDR is a full state protocol (i.e., maintains per-flow states) designed to manage resource reservation in the entire DiffServ domain. It is implemented only in the edge nodes of the domain and assures the interconnection between external protocols and the protocols inside the DiffServ domain. In this sense, the PDR may implement appropriate functions for performing admission control for incoming QoS-aware service requests based on resource availability, and mapping incoming requests to appropriate DSCP or PHB inside the domain. Moreover, it exploits signalling protocol (e.g., RSVP, NSIS [176], etc.), on per-flow basis, to request resource reservation for the PHB selected for each incoming flow, aiming at assuring a minimum bandwidth for admitted flows, implying that a flow is denied if there is not enough available bandwidth in a requested PHB. Further, the IP address of an ingress node on a path is notified to the corresponding egress node, allowing the egress nodes for notifying the ingresses about whether a requested reservation was successful on every node on a path. Also, egress nodes are responsible for notifying the ingresses about severe congestion situation that may occur (e.g., route changes due to link or node failures) such that ingress may react by stopping some affected flows and rejecting new requests according to the local control policies [86].

The PHR protocol is an extension to the PHB in Diffserv with support for resource reservation per PHB or traffic class and is implemented in each node within the Diffserv domain, thus the interior nodes only implement the PHR functions. To assist PDR in the management of resource within the domain, the signalling messages generated by PDR (e.g., QoS reservation request, refresh or release messages), usually on per-flow basis, are encapsulated in PHR messages at ingress nodes and sent down the network towards the corresponding egress nodes. Hence, the Resource Management in DiffServ (RMD) framework defines two PHR groups, namely the *Reservation-based* PHR group and the *Measurement-based Admission Control* (MBAC) PHR group.

Specified as the RMD On-demAnd (RODA) PHR protocol [177], the *Reservation-based* PHR group is a unicast edge-to-edge single DiffServ domain protocol, aiming at simplicity, cost-effectiveness and scalability. It enables each node, hosting the PHR, to perform admission control

for each QoS request based on parameter values conveyed in the reservation request signalling message and the available resources per traffic class. The admission is not based on typical measurement of data traffic load and available resource on the node, but on the parameters obtainable at each node since every interior node maintains reservation state per PHB (not per-flow) in terms of resource units.

The major limitations of the RMD approach can be summarized as in the following. The per-flow QoS reservation, refresh and the release signalling approach introduces not only undue control signalling and related processing overhead, but also a long session setup time. Moreover, each PHB is statically pre-configured on each node, leading to inflexible approach which would fail to satisfy highly dynamic and unpredictable resource demand inside a network. Besides, each node must perform admission control using the PHB reservation states and the pre-configured threshold parameters as inputs, which performance could be pushed to the network edge to improve scalability.

Besides, the work in [89] also addresses the resource under-utilization issues of aggregate RSVP proposals, by proposing a Scalable Reservation-based QoS (SRBQ) model for per-flow signalling-based resource reservation mechanisms. In SRBQ, it is considered that per-flow state, requiring memory, is no longer a problem in existing routers and efforts were concentrated on reducing the computational complexity and time associated with processing of every signalling message. In particular, signalling messages carry a label which provides each router with direct access to the corresponding flow reservation structures' address on memory, while an algorithm was developed for the efficient implementation of expiration timers used in the soft reservations.

To the best of our knowledge, existing aggregate resource over-reservation solutions showed serious limitations by wasting resources in attempt to reduce the control signalling overhead; the more the signalling overhead decreases, the lower the resource utilization efficiency goes, and operators would lose revenues. Moreover, periodic and on-demand probing in previous works increases design complexity, while inaccuracy and QoS violations are among other concerns. Further, none of them orchestrates synchronizations among edge nodes to leverage performance. Therefore, further investigations of aggregate resource over-reservations were strongly necessary to allow for system overall performance optimization with low control signalling load and high performance in terms of waste of resources or QoS violation. Besides the scope of BGRP, DARIS and SICAP is inter-domain, they are expected to generate instant signalling events, and thus the results of MARA show the superiority capability of over-reservation control against per-flow approaches. To the best of our knowledge, MARA is the most competitive and closely related to the

work carried out in this Thesis, and we will use it for comparison purposes in the analytical and simulations results.

2.8 *Network Survivability Control Proposals*

Network survivability has been one of the fundamental design goals to provide essential services in the presence of links/nodes failures and recover full service in a timely manner regardless of the scale, the magnitude, the duration and the type of failures [58], [59]. In general, application traffic might be divided into three categories requiring different levels of network survivability [57]. This includes: (1) *High-resilience-requirement traffic* (e.g., mission-critical, interactive tele-surgery, remote database transactions); (2) *Medium-resilience-requirement traffic* (e.g., standard VoIP and multimedia applications); (3) *Low-resilience-requirement traffic* (e.g., e-mail, File Transfer Protocol - FTP or standard World Wide Web - WWW). Thiran et al. [178] studied the differentiation of protection levels of two service classes as Fully Protected and Best Effort Protected. The Fully Protected is based on 1+1 or 1:1 protection at the WDM layer and offers a guarantee of fast survivability (e.g., 50ms or so). The Best Effort Protected is based on IP level restoration and relies on the amount of available spare capacity inside the network. A spare capacity allocation scheme consists in creating sufficient redundant capacity that is to be preallocated in a network and can be dedicated or shared [239]. In IP infrastructures, the widely deployed routing protocols (e.g., OSPF [4], Intermediate System to Intermediate System - IS-IS [60]) are able to reestablish connectivity after almost any failure of network elements. Table 2.1 summarizes standardized key tasks and related time constants as studied in [179], [180] upon failures to show OSPF convergence behavior divided into detection of failures, flooding of Links States Advertisements (LSAs), scheduling time of a Shortest Path First (SPF) calculation, and Forwarding Information Base (FIB) update.

Table 2.1. Main time constants in OSPF.

Name	Typically	Short Description.
T_{Hello}	10s	Interval between successive Hello packets.
T_{Dead}	4x T_{Hello}	Router Dead Interval (detect failure).
T_{spf}	*1-40 ms* [179]	SPF calculation (Depends on the node in the network).
$T_{spf\ delay}$	5s	Minimum time between LSA reception and start of SPF computation.
$T_{spf\ hold}$	10s	Minimum time between consecutive SPF computations.
T_{lsa}	0.6-1.1ms	Process LSA: check if LSA is new and update database.
$T_{lsa\ flood}$	33ms	LSA flooding time: process LSA, bundle LSAs and pacing time.
T_{fib}	100-300ms	Update the FIB: from end of LSA processing to end to end of new routes installation.

From Table 2.1, it is clear that IP networks typically take a few tens of seconds to converge, since all routers must update their FIB and synchronize with a consistent view of the network topology for packet forwarding to resume properly. Hence, they cannot satistify a large range of current and future applications (e.g., Telemedicine, IPTV, etc.), which only allow interruptions on the order of a few hundred milliseconds and less. Another issue with these protocols is the connectivity which may be restored through congested paths, even when there are other under utilized links in the network. Smita et al. [181] provide a classification of various proposals using dynamic and static link weights assignment to take traffic demands into account for SPF computation to minimize the congestion problems. However, complexity makes these solutions less suitable in dynamic scenarios. Moreover, tier-1 ISPs are opting for IP level restoration based on dynamic routing protocols, such as ISIS and OSPF [182], and careful design to couple IP restoration with adequate capacity provisioning is highly expected to achieve acceptable IP network survivability. The survivability complexity, flow re-routing and convergence time could be reduced if control could be pushed to network border, and forwarding tables of core nodes would not be required heavy updating effort upon failures.

Self-healing in ATM networks, as being the capability of a network to automatically recover itself from a failure of its components, has been intensively investigated to provide support for Virtual Circuits (VCs) route restoration [183]. The proposals can be classified into flooding type [184], [185] and backup Virtual Paths (VPs) type [186], [187], [188] with the goal to determine the optimal link capacities, link flows, and restoration routes in the network, while minimizing the total network cost [239]. The Capacity Allocation and Flow Assignment problem in ATM networks (CAFA problem) can be decomposed into optimal/near optimal network design problems, and the network survivability (network reliability evaluation) problem. In literature, these approaches have been broadly researched using linear programming, mixed integer linear programming, multi-commodity flow models, taking into account link/node failures scenarios and hop limit constraints [239], [189], [190]. However, none of them deploys aggregate resource over-reservation and control signalling overhead in finding resources for affected traffic remains a challenging issue. Taishi et al. [191] proposed priority control functions, splitting VPs into Guaranteed group and Best Effort group to achieve multi-reliability levels for VPs. By basing on protection techniques upon failures, the guaranteed VPs are first processed based on the VPs' reliability priority stored in each switching nodes' database. In case high reliability VPs cannot be recovered all using the existing backup VPs, it breaks the connectivity of the lower priority VPs to accommodate the higher ones as many as possible. While this solution provides valuable functions for priority-based flows re-routing, per-flow resource reservation and release signalling messages overhead pose scalability problems.

In MPLS enabled networks (e.g., MPLS-TP [192] and GMPLS [193]), system protection techniques are implemented by several architectures such as 1+1, 1:1, 1:n and m:n, which are usually referred to as Automatic Protection Switching in the context of Synchronous Digital Hierarchy/Synchronous Optical Networking (SDH/SONET). In 1:n architecture, a dedicated protection entity is shared by n working entities, and the m:n architecture is a generalization of the 1:n architecture. The shared protection methods improve system availability at smaller cost increases than the dedicated protection (1+1). Nonetheless, the recovery is delayed by signalling message exchanges in the management, control, or data planes to achieve the communication switchover after failures [194]. Moreover, the restoration approach is also used since local control policies can dictate paths setup priorities, such that recovery paths can preempt existing paths from lower priorities flows. In terms of cost-effectiveness, the shared-mesh restoration approach, which is the closest to our proposal, is known to allow multiple protecting paths, which may not have the same end points, to share common link and node resources, and thus assuring system availability with a more flexible resource-sharing and therefore less resource requirements. Hence, MPLS-enabled networks provide different grades of protection for different traffic classes within the same path based on the service requirements [192] using both protection and restoration recovery methods. A recovery time of 50 milliseconds has become the benchmark for emerging protection capabilities in MPLS, Dense Wave Division Multiplexing (DWDM), and the so-called wavelength routers (WR) [195]. Besides, the restoration time is much longer (hundreds of milliseconds), but outperforms the restoration time of tens of seconds in traditional IP re-routing. Moreover, existing network planning and the coordination of protection state after a recovery action are complex in shared-mesh restoration systems [196], [197]. In contrast to existing shared-mesh restoration approach, our proposal provides distributed CDPs at network border with good view of network topology and related resource conditions on real-time manner through aggregate resource over-reservation techniques. As such, the solution aims to prevent undue signalling overhead, while assuring differentiated QoS, which is mandatory to reduce session setup time for faster operations and flexible and flows re-routing.

Considering the modern communication networks as being constructed using a layered approach, significant research efforts have yielded many proposals of survivability in architectures such as IP/MPLS over WDM, ATM or SONET [198], [199], [200], [201]. Kayi et al [202] demonstrated that standard survivability metrics, such as the minimum cut and maximum disjoint paths [203], which have been widely used in characterizing the survivability properties of single-layer networks, lose much of their meaning in the context of cross-layer architecture. The authors propose two new survivability metrics, Min Cross Layer Cut to be the minimum number of physical failures that would disconnect logical topology, and Weighted Load Factor to quantify the

"impact" of each physical failure on the connectivity of multilayer networks. However, these approaches mostly target for theoretical and analytical studies. The work in [204] introduces an approach to achieve differentiated levels of service resilience using a dedicated protection scheme. It adjusts the size of working path areas protected by single backup paths according to the service class being protected, considering that the delay to restore broken connections depends on the length of backup paths. Nonetheless, this solution is strongly limited to single node failure scenarios and does not comply with very dynamic situations. Vadrevu et al [205] propose methodologies for integrated provisioning of wavelength and IP services with backup capacity sharing in IP-over-WDM networks. It uses Integer Linear Programming (ILP) formulation for ensuring connectivity of overlay IP topology (on top of WDM based physical topology) with sufficient capacity for re-routing all affected IP requests in the IP network under all single physical link failures. However, the focus on single link failure is a major limitation in this approach, since links and nodes may fail dynamically in real networks scenarios.

In [206], survivability is investigated in the context of network virtualization, where the impact of location-awareness is explored to address survivable network embedding using ILP solution for both active and backup traffic accommodation. While the authors address key challenges of survivable network embedding problem, the solution is too location-constrained. Survivability has also been broadly investigated in wireless technologies [207]. While the need for addressing multiple failures, in contrast to previous solutions, is gaining attention in the research community, existing studies are mostly yielding results from heuristic and probabilistic perspectives [208], [209], [210]. Hamza et al develop a p-cycle design solution in support for mixed-line rate (heterogeneous rate) optical networks aiming to reduce transponders and spare capacity requirements. Basically, the p-cycle concept [211] achieve very fast protection switching by routing pre-configured protection cycles over spare capacity in the network. It is more flexible than ring architectures where it can provide protection for both on-cycle links and the links (straddling) which are not directly on the cycle but whose end nodes are both on the p-cycle. However, it requires pre-configurations of the cycles. Therefore, survivability studies were deemed necessary to demonstrate support for stable operations of ACOR aware networks in presence of links/nodes failures.

From the previous described studies, there are no approaches addressing overprovisioning and QoS in the survivability approaches. Therefore, investigating survivability to assure stable operations, in presence of failures (e.g., of links and nodes), of the overprovisioning centric control mechanisms proposed in this Thesis, was deemed necessary.

2.9 *Context-Awareness*

Context-awareness has been broadly investigated within the European C-CAST project [173] with the main objective to evolve mobile multimedia multicasting to exploit the increasing integration of mobile devices with our everyday physical world and environment. C-CAST potentiates the use of sensor and smart devices environments (a.k.a. as smart space) to enable new personalization dimension to the global telecommunication market. A smart space could be any well-defined enclosed area such as a meeting room or school, or a well-defined open area such as a city square or national park. It typically comprises numerous heterogeneous sensors, smart devices and context information sinks, along with data servers with relevant (local public/environment) information, which interact with each other to provide enriched services and hence facilitate user immersive activities seamlessly. In literature, several definitions of context can be seen in [212]. Context may be any information that can be used to characterize the situation of entities (e.g., a person, a place, an object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Examples of context information from network user side are user geo-location, speed, direction, activity, battery power, device capability, transportation means, idle time, etc. From network perspective, context information may include congestion situation, resource usage, unpredictable re-routing, available network access points, QoS mapping statistics, and different QoS models models [213].

It is argued in [214] that a context-aware system must be able to sense and understand the answers to the types of questions generated from: who, what, when, where, and why; while context awareness is the state wherein a device or software program is aware of the environment and performs productive functions automatically. This implies that context-aware devices and programs are no longer passive entities waiting for instructions or commands, that is, they are alive and capable of intelligent behavior. Networks and services would exploit relevant context information to adapt their behavior to the changing circumstances in a very dynamic manner. The ubiquitous computing is also rapidly developing with mobile computing technologies, and there are several proposals which exploit sensor- and device-rich environment for personalized and pervasive human-centric computing as one can see in Projects Aura [215], Oxygen [216], BlueSpace [217] and Cooltown [218]. The same way, numerous proposals for service-oriented context-aware middleware have sprung in the community, Gaia Project [219], SOCAM [220], Context Toolkit [221], CoBrA [222], CMF [223]. More examples on context-aware applications can also be seen in [224], [225], [226], [214].

Network context-awareness is the ability of a system to use network context information to self-adapt or to the provision of services [227]. Hojin et al [228] use context server and Context-

Aware Messaging Server, and propose a “Join message free” context-based messaging services with multicast trees built in a top-down manner while they expect packet format to be more flexible in the future network. Roel et al [229] demonstrate context based flow classification and refer that currently, it is not possible to consider and classify flows comprehensively in terms of their wider context, simultaneously considering parameters that are internal and external to the flow itself, such as the kind of application that generated the flow, the characteristics of the device that will be consuming the flow, the activities of the user who generated the flow, etc.

This is a fundamental concept in the work developed in this Thesis since network entities (e.g., CDPs) are expected to dynmically learn from one another’s context information (e.g., resource usage, congestion situation, etc.) for a proper self-organization and self-control.

2.10 *Summary*

This chapter introduced some background for the work achieved in this Thesis with focus on the most relevant related work. It includes the main IETF standardized QoS and resource control architectures such as the InServ, DiffServ and MPLS, and depicts the existing admission control models (the active measurement-based, the passive measurement based, and the parameter-based). Besides, we described IP multicast technology, presented signalling protocols, and introduced the architecture and requirements of NGNs along with the three major standards for resources and admission control (e.g., IMS, RACF and RACS). Due to the similarity between these standards, their overall functionalities and the standardized control operations in terms of session establishment and the related QoS-aware transport within the NGNs were illustrated using the IMS-enabled network, and the key building functional blocks of the RACS were described to ease the understanding of the functionalities and the similarity. Networking control paradigms and proposals for centralization and decentralization have been studied to highlight the advantages and disadvantages of each approach. Moreover, we explored scalable resources and admission control proposals along with key frameworks which combine the QoS guarantees benefits of IntServ with the scalability and the flexibility features of DiffServ. Further, we also surveyed the most relevant proposals on network survivability control, and described context-awareness and its relevance in current and future networking scenarios, since these approaches are integral parts of the research work carried out in this Thesis.

Chapter 3

Overprovisioning in Class-Based Networks: COR and ECOR

As we studied in the previous chapters, aggregate resource overprovisioning approach envisions reserving to each CoS more bandwidth than currently required, also known as over-reservation, such that admission control can process several session requests (admission, release or readjustment) with less signalling overhead than in per-flow mechanisms. However, this requires that each admission decision inside a network is based on a good view of the network topology and the related link resource statuses in every CoS, *on real-time basis*, to avoid *QoS violation* and *waste of resources*. Moreover, appropriate algorithms can exploit the updated resources information and dynamically distribute residual resources (over-reserved but unused) among CoSs on network interfaces in order to prevent *CoS starvation* [230]. In the context of over-reservation in this Thesis, *QoS violation* refers to the situation where the number of sessions admitted in a CoS exceeds the maximum allowable, which affects not only the newly admitted sessions, but all existing ones in the CoS. *Waste of resources* or network under-utilization occurs when inefficient admission decision rejects an authorized demand for a certain amount of resources which is currently available in the network, thus increasing session blocking probability unnecessarily.

To the best of our knowledge, existing overprovisioning solutions mostly acquire network resource capabilities based on periodic and on-demand probing techniques, which increases signalling and the related processing overhead when attempting to decrease the probing period to improve information consistency for admission decisions. The recently proposed MARA [125] approach distinguishes itself from earlier state-of-the-art over-reservation solutions by implementing specific functions to dynamically deal with residual resources among CoSs.

Nonetheless, the solution relies on the traditional periodic probing schemes and avoids reserving too much surplus of resources to each CoS in attempt to alleviate the issue of QoS violation and waste of resources. Moreover, the resource readjustment functions of MARA fail to properly distribute residual resources among CoSs. Hence, although very interesting in terms of dynamic resources over-reservation control in the NGN, the approach shows serious limitations in terms of CoS starvation, waste of resources and QoS violations, which attracted our attention for further investigation.

This chapter proposes two aggregate resource over-reservation algorithms, the COR and the ECOR. The main objective of COR is to improve MARA's resource readjustment algorithm. To this end, COR uses MARA's function to compute surplus of reservation for CoSs, and introduces key functions to efficiently deal with residual reservations among CoSs in a way to prevent CoS starvation, waste of bandwidth or unnecessary session blocking. This way, COR and MARA cannot reserve too much surplus per CoS and clearly fail to allow for optimizing signalling and the related processing overhead minimization. This motivates our efforts to design the ECOR, which allows for reserving as much resources as possible for each CoS while being able to efficiently negotiate the residual bandwidth to prevent CoS starvation or waste of resources as well. Further, we provide an analytical model for analyzing the impact of a set of parameters (e.g., session lifetime, bandwidth usage, interface capacity, etc.) on aggregate resource over-reservation control performance in general, in terms of signalling overhead minimization, waste of resources, and therefore, the increase of session blocking.

We propose the ACA centralized architecture in which a central CDP is enabled to maintain its underlying network topology and the related links resource conditions on real-time. For this purpose, ACA creates multiple edge-to-edges multicast trees and records the lists of outgoing interfaces of each tree in its local database called *Network Context Information Base* (NetCIB). Hence, whenever the CDP admits, releases, or readjusts a session requirements in a CoS on a tree, it automatically updates the resource utilization statistics for each outgoing interface that composes the concerned tree in its NetCIB, thus keeping a good knowledge of resource statistics in each CoS on every interface. The data sessions enjoy the QoS destined to them on trees and the CDP can assure consistency of its local database information while avoiding signalling the trees. Hence, we implement COR, ECOR and MARA as use cases in the ACA architecture, and evaluate its performance analytically and through simulation. The results that we obtain demonstrate that it is possible to significantly reduce QoS reservation control signalling, and therefore, the related processing overhead to achieve scalability using the concept of aggregate resource over-reservation without suffering QoS violations, CoS starvation, waste of resources or unnecessary increase of session blocking probability.

This chapter is organized as follows. Section 3.1 describes the COR algorithm and section 3.2 presents the ECOR scheme. In section 3.3, we provide a generic analytical model for resources over-reservation using COR, ECOR and MARA as use cases. Section 3.4 describes the ACA architecture and section 3.5 presents the performance analysis for both analytical and simulation results. Finally, section 3.6 concludes the chapter.

3.1 *COR Scheme*

The COR is a resource computational procedure which provides a set of functions that allow for dynamically defining bandwidth over-reservation parameters among various CoSs on network interfaces upon need. The main objective of COR is to improve the performance of MARA, mainly by efficiently reusing residual resources on network interfaces such that the over-reserved but unused bandwidth can always be dynamically provisioned to the CoSs which need it, in a way to prevent resource starvation and waste, which is very important to avoid increasing session blocking probability unnecessarily. In general, the proposed functions are classified into two categories: 1) Reservations Initialization functions which provision each CoS on an interface with a certain amount of bandwidth, considering that the interface is not being used currently (e.g., at system bootstrapping); 2) Reservations Parameters Readjustment functions which allow for redefining reservation parameters on an interface dynamically during system running-time. Hence, these functions can be used in network admission control mechanisms to dynamically compute bandwidth parameters to be enforced through schedulers in network upon need [64], [65] to provide QoS. The way COR initializes resources reservation parameters and readjusts them on demand for CoSs implemented on a given network interface is detailed in subsection 3.1.1 and subsection 3.1.2 respectively.

3.1.1 **Reservations Parameters Initialization Functions.**

In order to perform a flexible resource distribution among CoSs for a given network interface, COR allows for assigning a weight w_i to each service CoS_i on the interface. The weights can be specified by the network administrator (e.g., taking resource needs of CoSs into account). Moreover, COR uses the reservation parameters of MARA, such as, a parameter $R_{BW}(i)$ which represents the amount of bandwidth to be reserved for each CoS_i, a parameter $\chi_{BW}(i)$ which stands for the maximum (threshold) that $R_{BW}(i)$ must not exceed in the CoS_i, and a global initialization factor *Index* (e.g., 1/2, 1/3, 1/4, etc.) as being a fraction of the threshold $\chi_{BW}(i)$, which prevents from reserving the overall resources of an interface at system initialization phase. To facilitate the

understanding, let's consider that a network interface I_e, implementing one control CoS and k service CoSs has a total capacity C, and a fixed amount of bandwidth b is reserved for the control CoS. Hence, the initial reservation parameters computation process for I_e, as illustrated using *steps 1, 2 and 3* in Figure 3.1, is detailed in the following. For the sake of simplicity, COR assigns a weight to each CoS_i ($1 \le i \le k$) as:

$$w_i = \frac{1}{k} \tag{3.1}$$

Then, the reservation $R_{BW}(i, I_e)$ and the threshold $\chi_{BW}(i, I_e)$ of each CoS_i are defined based on its weight using the following functions:

$$\chi_{BW}(i, I_e) = w_i * (C - b) \tag{3.2}$$

$$R_{BW}(i, I_e) = Index * \chi_{BW}(i, I_e) \tag{3.3}$$

3.1.2 System Operating Functions

As we referred in Chapter 2 when resources are over-reserved and a network is running, an admission decision point (e.g., QoS Broker, ingress router, etc.) can process several session requests (e.g., session setup, release or traffic requirements readjustment) without issuing QoS reservation signalling into the network, thus reducing signalling and related processing overhead, in contrast to per-flow approaches. However, when the reservation of a requested CoS_j ($1 \le j \le k$) is insufficient in all candidate trees to admit a new request r_j ($R_{BW}(j, I_e) < U_{BW}(j, I_e) + r_j$), where $U_{BW}(j, I_e)$ is the used bandwidth in CoS_j, the COR may be triggered for reservation parameters readjustment among CoSs on relevant interfaces inside the network. It is very important to notice that the decision about which interface(s) should be processed is taken by the concerned admission decision point. Hence, the way COR computes reservation readjustment parameters on an interface I_e is summarised in Figure 3.1 (steps 4 through 16) and detailed in the following.

First, the total amount of unused bandwidth Δ_T on I_e is computed in step 4 of Figure 3.1 as:

$$\Delta_T(I_e) = \sum_{i=1}^{k} \left[\chi_{BW}(i, I_e) - U_{BW}(i, I_e) \right] \tag{3.4}$$

When $\Delta_T(I_e)$ is insufficient for the incoming request r_j, COR reports that the interface I_e is considered as in step 16. Otherwise, COR proceeds with the process as described in the following.

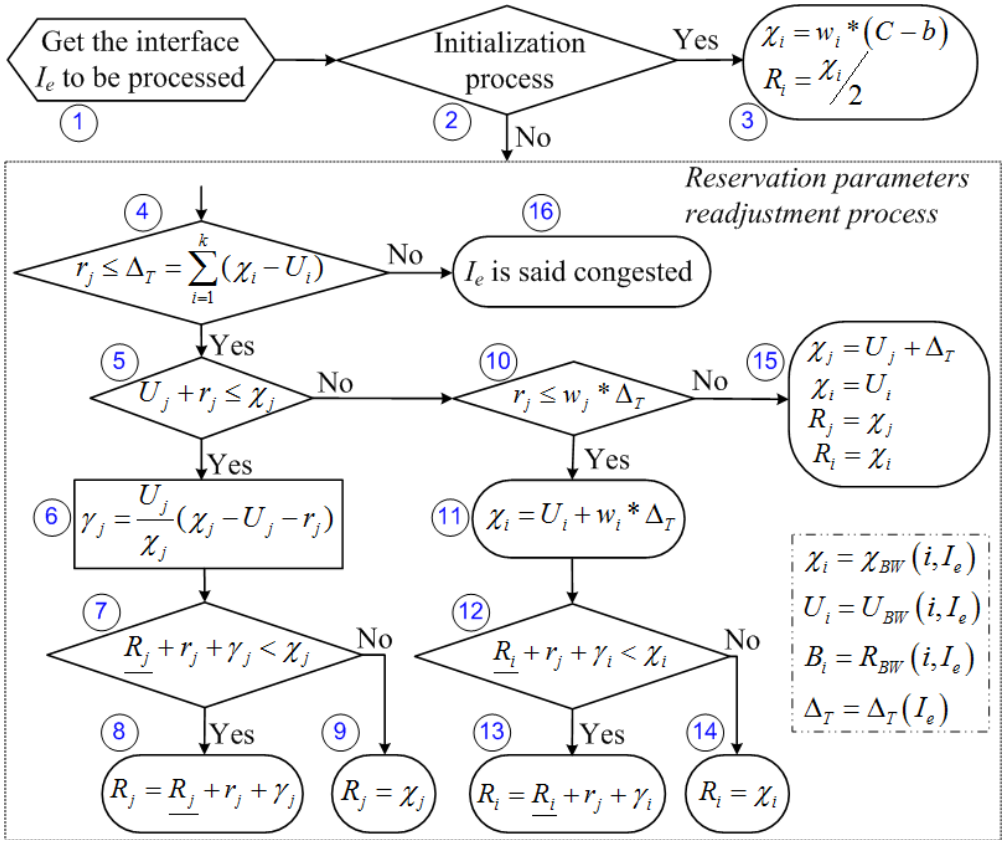


Figure 3.1. COR algorithm flow chart.

When the maximum reservation threshold of CoS_j is not exhausted ($\chi_{BW}(j, I_e) \ge U_{BW}(j, I_e) + r_j$), new amount of over-reservation bandwidth γ_j is computed *(step 6)* using the same function of MARA [125] as:

$$\gamma_j = \frac{U_{BW}(j, I_e)}{\chi_{BW}(j, I_e)} * \left[\chi_{BW}(j, I_e) - U_{BW}(j, I_e) - r_j \right] \tag{3.5}$$

Then, new reservation parameter $R_{BW}(j, I_e)$ is obtained for the congested CoS_j using the following functions (*steps 7-8 or 7-9*), depending on the resource conditions on the interface:

IF $\underline{R_{BW}(j, I_e)} + r_j + \gamma_j < \chi_{BW}(j, I_e)$, update:

$$R_{BW}(j, I_e) = \underline{R_{BW}(j, I_e)} + r_j + \gamma_j \tag{3.6}$$

Otherwise

$$R_{BW}(j, I_e) = \chi_{BW}(j, I_e) \tag{3.7}$$

where $\underline{R_{BW}(j, I_e)}$ is the old reservation to be updated for the CoS_j.

However, when the maximum reservation threshold of CoS_j is exhausted ($\chi_{BW}(j, I_e) < U_{BW}(j, I_e) + r_j$), MARA reduces unused bandwidth from all other CoS_i with ($i \neq j$) to decongest CoS_j, which functions show serious limitation in terms of waste of resources as we detail later in section 3.3. In contrast, COR redistributes the total unused bandwidth among all service CoSs on the interface based on the weights assigned to the CoSs, thus aiming at achieving a more balanced resource redistribution than MARA. To this end, COR first computes a certain amount of unused bandwidth η_i that can be allocated to each CoS_i according to its weight as:

$$\eta_i = w_i * \Delta_T(I_e) \tag{3.8}$$

Hence, according to the output of equation (3.8), one of the following three situations will occur:

- If the amount of bandwidth η_j of CoS_j is sufficient *(step 10)* to admit the request ($r_j \leq \eta_j$), the maximum reservation threshold $\chi_{BW}(i, I_e)$ of every CoS_i on the interface is readjusted in step 11 by:

$$\chi_{BW}(i, I_e) = U_{BW}(i, I_e) + w_i * \Delta_T(I_e) \tag{3.9}$$

Consequently, the new reservation parameter $R_{BW}(i, I_e)$ of each CoS_i ($1 \leq i \leq k$) on the interface is readjusted (steps 12-13 or 12-14), depending on the amount of the available resources on the interface as follows:

IF ($\underline{R_{BW}(i, I_e)} + r_j + \gamma_i < \chi_{BW}(i, I_e)$), compute:

$$R_{BW}(i, I_e) = \underline{R_{BW}(i, I_e)} + r_j + \gamma_i \tag{3.10}$$

Otherwise,

$$R_{BW}(i, I_e) = \chi_{BW}(i, I_e) \tag{3.11}$$

Where $\underline{R_{BW}(i, I_e)}$ is the old reservation to be updated for CoS_i.

- However, if the output η_j of CoS_j in equation (3.8) is insufficient to admit the request ($r_j > \eta_j$), the amount $\Delta_T(I_e)$ obtained in equation (3.4) is allocated to the CoS_j (step 15) to allow for admitting the request successfully, since the condition ($r_j \leq \Delta_T(I_e)$) is guaranteed in step 4. This means that COR readjustment functions always allow for admission provided that the total unused resource on an interface is enough for the incoming demand which has invoked the computations. Thus, COR effectively avoids CoS starvation, unnecessary waste of bandwidth and unnecessary

increase of session blocking probability. Hence, new threshold and reservation parameters of CoS_j are computed as:

$$\begin{aligned} \chi_{BW}(j, I_e) &= U_{BW}(j, I_e) + \Delta_T(I_e) \\ R_{BW}(j, I_e) &= U_{BW}(j, I_e) + \Delta_T(I_e) \end{aligned} \tag{3.12}$$

Consequently, the reservation and threshold parameters of each of the remaining CoS_i with ($i \neq j$) are updated using the following functions:

$$\begin{aligned} \chi_{BW}(i, I_e) &= U_{BW}(i, I_e) \\ R_{BW}(i, I_e) &= U_{BW}(i, I_e) \end{aligned} \tag{3.13}$$

This way, COR is able to define new reservation parameters for CoSs on any interface inside a network. Hence, it becomes clear that a session admission decision point can exploit COR to dynamically regulate resource allocation among CoSs on network interfaces in support for scalable QoS provisioning without jeopardizing performance in terms of CoS starvation and waste of resources. In master-client such as in PDP-PEP mode, the specified reservations parameters by a PDP can be encapsulated and conveyed in appropriate control signalling messages to relevant nodes (PEPs) inside a network to carry out the reservations decisions enforcement to assure that traffic will receive the QoS expected. As we referred, admission decision and the related signalling control procedures are provided in section 3.4.

3.2 *ECOR Scheme*

COR demonstrates significant contributions to improve the performance of MARA by providing appropriate functions to efficiently control over-reservation parameters across a network without incurring CoS starvation and waste of resources. However, the solution is too dependent on MARA. In particular, it uses the same function (please see equation (3.5)) as MARA to compute the amount of surplus to reserve for each CoS, which function strickly prevents from reserving too much resources to CoSs. As a consequence, both COR and MARA clearly fail to allow for optimizing the signalling and related processing overhead minimization. Moreover, the algorithm involves computational steps and parameters and thus raises scalability problems.

Bearing this in mind, we propose the ECOR which allows for reserving to each CoS on an interface as much resources as there are available on the interface, while efficiently dealing with the residual reservations in a way that prevents CoS starvation and waste of resources. This way, ECOR focuses on allowing the optimization of control overhead, while incorporating all the benefits of COR by avoiding resource starvation and waste. Like COR, it is important to mention that ECOR is not an admission control mechanism. Rather, it is a scheme that can be exploited by

an admission control system to dynamically define and readjust reservation parameters for CoSs on any network interface upon need for support to QoS provisioning. Hence, the functions introduced by ECOR are also classified into Reservations Initialization functions and Reservations Parameters Readjustment functions as we detail in sections 3.2.1 and 3.2.2 respectively.

3.2.1 System Initialization Functions

In order to allow for flexible resource management among CoSs, ECOR assigns a weight w_i to each service CoS_i like in COR using the equation (3.1). However, ECOR control bandwidth distribution among CoSs on an interface is based on the amount of the available resources and the weights assigned to the CoSs on the interface without defining any reservation threshold, in contrast to COR and MARA. Moreover, ECOR removes the use of a global factor called Index in COR which inherited it from MARA, since the primary goal of ECOR is to allow for reserving as much resources as possible, and relies on efficient redistribution to improve the resource sharing in a network. As such, ECOR initializes (steps 1-2-3 in Figure 3.2) reservation parameters for each service CoS_i on an interface based on the weight of each CoS_i using the following function.

$$R_{BW}\left(i, I_e\right) = w_i * \left(C - b\right) \tag{3.14}$$

3.2.2 System Operating Functions

When an admission control mechanism, which implements ECOR, receives a session request r_j in a CoS_j and realizes that the available reservation in the CoS_j is insufficient for admission, ECOR should be invoked to obtain possible new reservation parameters for reconfigurations on relevant interfaces inside the network so that the request may be admitted. Hence, the way ECOR defines new reservation parameters for CoSs on an interface is detailed in the following.

First, the total unused bandwidth Δ_T on the interface is obtained as:

$$\Delta_T\left(I_e\right) = \sum_{i=1}^{k}\left[R_{BW}\left(i, I_e\right) - U_{BW}\left(i, I_e\right)\right] \tag{3.15}$$

where $R_{BW}\left(i, I_e\right)$ and $U_{BW}\left(i, I_e\right)$ are respectively the reservation and used bandwidth of each CoS_i, with $1 \le i \le k$ on the interface I_e. Hence, in case Δ_T is smaller that the demand r_j (in step 4 in Figure 3.2), ECOR reports that the interface I_e is congested as in *step 8*. Otherwise, ECOR defines new parameters for each CoS_i on the interface as in the following. First, the request r_j is compared with the weighted portion of the total unused bandwidth on the interface as in *step 5*. Hence, if this

portion is sufficient for the request to the CoS_j ($r_j \le w_j * \Delta_T(I_e)$), new reservation parameters $R_{BW}(i, I_e)$ are defined for each CoS_i on the interface in step 6 as:

$$R_{BW}(i, I_e) = U_{BW}(i, I_e) + w_i * \Delta_T(I_e) \tag{3.16}$$

However, in case the weighted portion is insufficient, ECOR allows for efficiently utilizing the residual reservations to avoid waste of bandwidth, CoS starvation or unnecessary service requests blocking. In particular, it defines new reservations $B_{BW}(j, I_e)$ for the requested CoS_j and $B_{BW}(i, I_e)$ for each of the remaining CoS_i on the interface as in step 7 using the following expressions:

$$\begin{aligned} R_{BW}(j, I_e) &= U_{BW}(j, I_e) + r_j + w_j * (\Delta_T(I_e) - r_j) \\ R_{BW}(i, I_e) &= U_{BW}(i, I_e) + w_i * (\Delta_T(I_e) - r_j), \quad 1 \le i \le k, \ (i \ne j) \end{aligned} \tag{3.17}$$

From equation (3.17), one can see that the reservations $R_{BW}(j, I_e)$ of the CoS_j (demanded CoS) are updated by granting the requested amount r_j from the total unused resources on the interfaces. Then, the remained unused resources are distributed among all the CoS_i based on the weight of each CoS. This proves that the resource readjustment functions of ECOR would always succeed to allow for admission through an interface as long as the total unused bandwidth on the interface is greater than the demand, thus preventing CoS starvation and waste of resources, and therefore avoids unnecessary increase of session blocking probability. Moreover, ECOR requires less control parameters and procedures (please see Figure 3.1 and Figure 3.2) which is of paramount importance to scale.

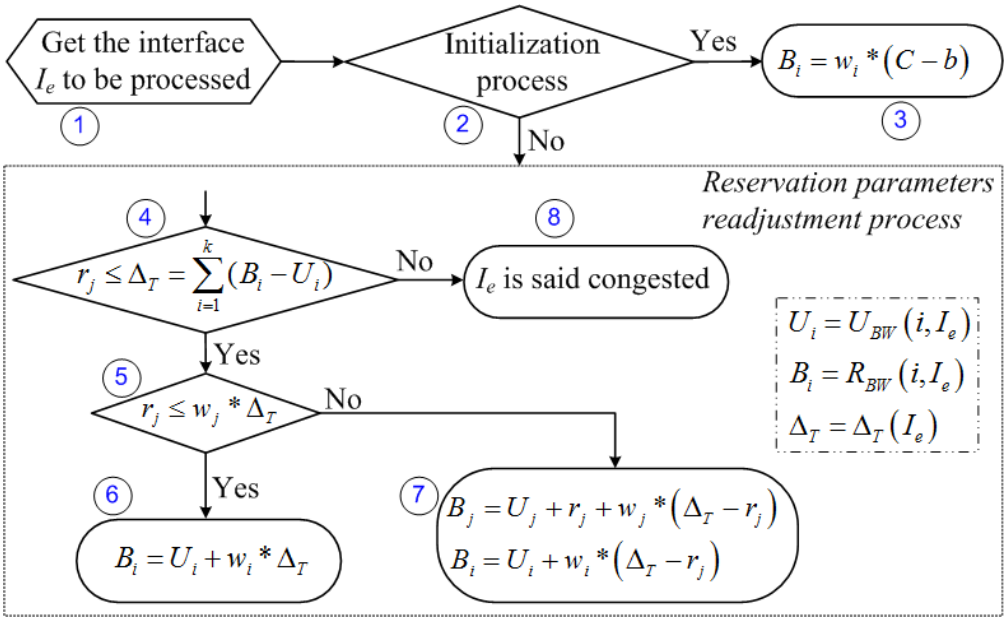


Figure 3.2. ECOR algorithm flow chart.

To get the insight of COR, ECOR and MARA, as well as the impact of various control parameters on their performance in terms of signalling overhead minimization and waste of resources, the subsequent section provides an analytical study to assess over-reservations schemes in general.

3.3 *Analytical Model for Resource Over-Reservation Schemes*

As we studied in Chapter 2, aggregate bandwidth over-reservation is a promising approach to reduce QoS reservation signalling and the related overhead. Therefore, this section provides analytical studies to facilitate a good understanding of major benefits and challenges when referring to aggregate bandwidth over-reservation control. Our study bases on three major use cases such as COR, ECOR and the competing state-of-the-art MARA's solution. Knowing that QoS reservation signalling is usually triggered by reservation exhaustion on bottleneck interfaces inside a network, we use Figure 3.3 to illustrate a bottleneck outgoing interface I_b of a network node **A** towards a node **B** on a communication path.

3.3.1 Over-Reservation Model

Let's consider that the bottleneck interface I_b, as in Figure 3.3, has a capacity C and implements one control CoS with a certain dedicated bandwidth reservation b, and k service CoSs (e.g., EF, AF, BE, etc.) to which the remained capacity $(C-b)$ is destined for sessions transport.

Hence, the amount of bandwidth surplus $R_{BW}(i, I_b)$ that an over-reservation algorithm (e.g., COR, ECOR or MARA) computes for a given CoS_i with ($1 \le i \le k$) on the interface can be

expressed as:

$$R_{BW}\left(i,I_b\right)=f\left(\Psi\right) \tag{3.18}$$

where, ψ stands for the bandwidth over-reservation computational procedure (e.g., equations) of the algorithm in use.

For the sake of simplicity, we assume that session requests to a CoS_i through the interface I_b are Poisson processes with rate λ_i and the mean amount of bandwidth demand of each session in a CoS is $\bar{r}$. This way, the total number n of sessions that a surplus of over-reserved bandwidth $R_{BW}\left(i,I_b\right)$ of a CoS_i (obtained in equation (3.18)) can accommodate simultaneously on the interface I_b without requiring a reservation readjustment signalling event can be obtained as:

$$n=\left\lfloor \frac{R_{BW}\left(i,I_b\right)}{r_i} \right\rfloor \tag{3.19}$$

Then, it is assumed that a session's lifetime τ is exponentially distributed such that $\tau = 1/\mu$, where μ is a real number (service rate) in sessions per time unit. This means that the longer a session lifetime is, the smaller the related service rate is.

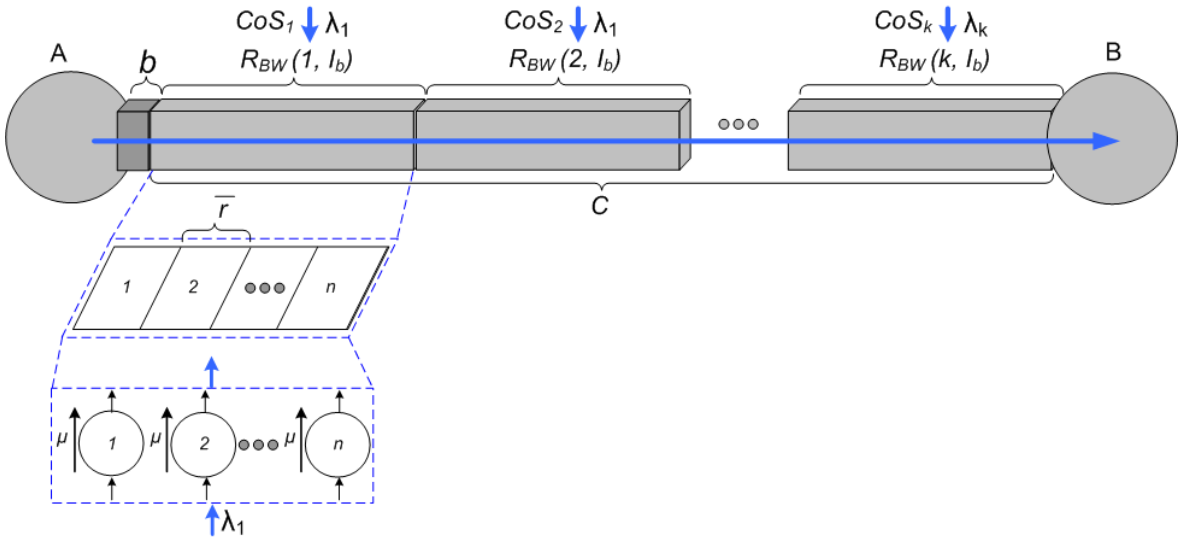


Figure 3.3. Bottleneck interface model.

This way, it turns out that an over-reserved bandwidth $R_{BW}\left(i,I_b\right)$ of a CoS_i is characterised by:

- n possible sessions slots of a mean bandwidth $\bar{r}$;
- Session requests to a CoS_i are Poisson processes with rate λ_i;
- A session's lifetime is exponentially distributed with mean τ;

- Session requests arrival process is independent of the session lifetime;
- A given amount of over-reserved bandwidth $R_{BW}\left(i, I_b\right)$ can only accommodate *n* sessions simultaneously.

The over-reservation readjustment control approach is then modelled as an M/M/n/n queuing system as depicted in Figure 3.3, where the first *n* is the number of slots reserved for sessions while the second *n* is the maximum number of sessions that can be running at a time. Thus, the probability P_i that the over-reserved bandwidth of a CoS_i exhausts on the bottleneck outgoing interface of a path to trigger reservation readjustment event is the probability that an incoming service request finds all *n* over-reserved sessions slots occupied. Hence, the probability P_i can be obtained using Erlang B formula as:

$$P_i = \frac{\left(\frac{\lambda_i}{\mu}\right)^n * \frac{1}{n!}}{\sum_{\alpha=0}^{n}\left(\frac{\lambda_i}{\mu}\right)^\alpha * \frac{1}{\alpha!}} \tag{3.20}$$

where, *n* is obtained in equation (3.19) and α is an integer number.

From the equations (3.18), (3.19), (3.20), it becomes clear that the frequency of over-reservation readjustment events depends on several parameters. These parameters include the mean bandwidth $\bar{r}$ allocated to each session, session requests arrival rate λ_i to the CoS_i, session mean lifetime τ, the number of CoSs implemented on interfaces and network interface's capacity *C* or interface resource utilization level. This means that over-reservation algorithms must be carefully designed to achieve improved performance as we further study in the subsequent subsection based on COR, ECOR and MARA.

3.3.2 **Over-Reservation Algorithm Model**

In order to model the behaviors of the over-reservation algorithms and ease further understanding on over-reservation approach, especially during a dynamic system operation phase, the over-reservation computation functions of COR, ECOR and MARA are studied under different resource utilization conditions such as *Low* and *High utilization levels* separately. This is important as we take into account the general behaviors of each of the algorithms being studied. Hence, by knowing *b* and the capacity *C*, the total amount of used bandwidth $U_{BW}^{Total}\left(I_b\right)$ on the interface I_b can be obtained as:

$$U_{BW}^{Total}(I_b) = \bar{r} * \left(\left\lfloor \frac{C-b}{\bar{r}} \right\rfloor - Q \right), \quad 0 \le Q \le \left\lfloor \frac{C-b}{\bar{r}} \right\rfloor \tag{3.21}$$

where, $\bar{r}$ is the mean bandwidth required by each session on the interface, Q is the integer number of unused session slots (free resource slots) on the interface I_b, and $\lfloor \bullet \rfloor$ is the nearest integer value which is smaller than or equal to "$\bullet$". Hence, given the parameters $\bar{r}$, C and b, an interface utilization level (*low* or *high*) can be inferred through Q: the higher the integer Q is, the lower the interface resource utilization level is.

Based on the equation (3.21), the total unused resources $\Delta_T(I_b)$ on the interface I_b are obtained by:

$$\Delta_T(I_b) \approx \bar{r} * Q \tag{3.22}$$

Then, the bandwidth surplus computation functions of each algorithm, under low or high resource utilization conditions, considering that the process is triggered by a requested CoS_j, are provided in the following.

Low Utilization Phase: traffic density is considered Low as long as the following condition (3.23) is fulfilled.

$$\bar{r} \le w_j * \Delta_T(I_b) \Rightarrow \frac{Q}{k} \ge 1 \tag{3.23}$$

Then, the computation functions of each algorithm can be summarized as in the following.

ECOR: it computes a new surplus $R_{BW}(i, I_b)$ for each CoS_i based on the equation (3.16) as:

$$R_{BW}(i, I_b) = w_i * \Delta_T(I_b) \approx \frac{\bar{r} * Q}{k} \tag{3.24}$$

MARA: First, MARA computes a bandwidth index (*B_Idxi*) of each CoS_i $1 \le i \le k$, $(i \ne j)$, except for the requested CoS_j and defines a threshold index (*Th_Idxi*) of each CoS_i. It computes a certain amount of bandwidth (*Brl_Xi*) that it removes from the threshold $\chi_{BW}(i, I_b)$ of the CoS_i to increase the threshold of the congested CoS_j where:

$$Brl\_\chi_i = \left(\frac{B\_Idx_i + Th\_Idx_i}{2} \right) * \left(\chi_{BW}(i, I_b) - Bref_i \right) \tag{3.25}$$

$Bref_i$ is a bandwidth reference, which is either the bandwidth currently reserved for the CoS_i or the Committed Reservation threshold $CRth_i$ of the CoS_i (if $u_i(I_b)$ is lower than the *CRthi*). This way,

MARA uses the equation (3.25) to remove a certain amount of bandwidth from each of other CoS_i and uses the sum to increase the threshold $\chi_{BW}(j, I_b)$ of the requested CoS_j. Assuring that it removes no more than the unused resource on each CoS_i, it computes a new amount γ_j (non negative) of bandwidth for the CoS_j using equation (3.5) and uses the output to compute a new surplus of over-reservation $R_{BW}(j, I_b)$ within the threshold $\chi_{BW}(j, I_b)$ for the CoS_j as:

$$R_{BW}(j, I_b) = \gamma_j + \overline{r} \tag{3.26}$$

COR: it readjusts the threshold $\chi_{BW}(i, I_b)$ of every CoS_i with $1 \le i \le k$ as:

$$\chi_{BW}(i, I_b) = U_{BW}(i, I_b) + w_i * \Delta_T(I_b) \approx U_{BW}(i, I_b) + \frac{\overline{r} * Q * w_i}{k} \tag{3.27}$$

where $U_{BW}(i, I_b)$ is the used bandwidth in the CoS_i.

Then, it computes new surplus of bandwidth over-reservation $R_{BW}(j, I_b)$ for the requested CoS_j like MARA by using the equation (3.26).

High Utilization Phase: with respect to the algorithms, resource utilization level is said high on an interface whenever the following condition is met:

$$w_j * \Delta_T(I_b) < \overline{r} \le \Delta_T(I_b) \Rightarrow \frac{Q}{k} < 1 \le Q \tag{3.28}$$

In this case, each scheme computes the over-reservations as in the following.

ECOR: it computes new surplus of bandwidth $R_{BW}(j, I_b)$ for the congested CoS_j on an interface I_b using the equation (3.17) as:

$$R_{BW}(j, I_b) = \overline{r} + w_j * (\Delta_T(I_b) - \overline{r}) \approx \overline{r} * \frac{Q + k - 1}{k} \tag{3.29}$$

Besides the requested CoS_j, it distributes the remained unused resource to each of the other CoS_i in an attempt to reduce resource exhaustion probability where:

$$R_{BW}(i, I_b) = w_i * (\Delta_T(I_b) - \overline{r}) \approx \overline{r} * \frac{Q - 1}{k} \tag{3.30}$$

MARA: In such case, MARA increases the threshold of the requested CoS_j by removing a certain amount of resource from each other CoS_i, using the same functions in the equation (3.25).

COR: In this situation, COR readjusts the thresholds $\chi_{BW}(j, I_b)$ of the requested CoS_j and

$\chi_{BW}(i, I_b)$ of the other CoS_i as:

$$\begin{aligned} \chi_{BW}(j, I_b) &= U_{BW}(j, I_b) + \Delta_T(I_b) \approx U_{BW}(j, I_b) + \bar{r} * Q \\ \chi_{BW}(i, I_b) &= U_{BW}(i, I_b) \end{aligned} \tag{3.31}$$

Then, it defines new surplus of bandwidth $R_{BW}(j, I_b)$ and $R_{BW}(i, I_b)$ for the requested CoS_j and each of the other CoS_i as:

$$\begin{aligned} R_{BW}(j, I_b) &= \Delta_T(I_b) \approx \bar{r} * Q \\ R_{BW}(i, I_b) &= 0 \end{aligned} \tag{3.32}$$

When compared with MARA, the readjustment method of COR exploits weights of CoSs for better redistribution of resources.

3.4 *ACA Control Mechanism*

The existing networks mostly implement centralized solutions [173], which require scalable support for their control to facilitate service creation. As we observed in the previous and in this chapter, aggregate resource over-reservation is promising for scalability. The major challenge is that the approach strongly requires a good knowledge of network topology and the related resource utilization statistics in *real-time* to prevent performance degradation in terms of QoS violations and waste of resources. In sections 3.1 and 3.2, we introduced COR and ECOR respectively, to address the issue of efficient management of residual reservations to overcome resource starvation and related unnecessary session blocking. Nonetheless, these algorithms, as well as any over-reservation algorithm (e.g., MARA) need adequate network support to obtrain accurate resources statistics to effectively attain results.

Therefore, we propose the ACA in which a central network CDP is enabled to maintain its underlying network topology and the related links resources statistics on real-time basis. The ACA operations are based on two fundamental principles: 1) Explicit Routes Implementation, deployed by means of multicast trees to assure that the packets of a session mapped to a tree are forced to follow the desired tree; 2) Real-Time Topology and Resource Statistics Update, which is achieved by recording the lists of outgoing interfaces that compose every tree created inside the network. Moreover, whenever the CDP admits, releases or readjusts a session in a CoS on a tree, it automatically updates the resources statistics in the CoS on each of the outgoing interfaces on that tree in its local NetCIB database accordingly. The rest of this section describes the ACA's architecture and the proposed functionalities using the QoS and admission control reference model of ETSI/TISPAN introduced in subsection 2.5.2 in Chapter 2.

3.4.1 **ACA Control Architecture**

In ACA, a central CDP is responsible for the overall control of the underlying network infrastructure as illustrated with Figure 3.4, encompassing a CDP, 5 Border Routers (BRs) and 4 Core nodes (Cs). Based on TISPAN reference architecture and taking network nodes' locations (e.g., border, core, etc.) and functionalities into account, ACA defines three control agents that various nodes can implement for proper interactions and control in the network. These agents include: (1) ACA-Full agent (ACA-F) which is a state-full agent specified for the CDP and therefore embeds the SPDF and RACF functions; (2) ACA-Border agent (ACA-B), a semi-full state agent specified for the BRs by including the BGF, RCEF and BTF functions; (3) ACA-Light agent (ACA-L) as a light weight agent defined with basic functionalities of RCEF and BTF (e.g., packet forwarding, QoS enforcement, etc.) as required in all nodes which handle traffic. Further details on these agents and their operations are provided in subsequent sub-sections.

3.4.1.1 ACA-Full Agent

Designed for the CDP as a state-full agent, ACA-F implements the SPDF functions to enable the CDP for receiving session requests (e.g., session setup, release, session readjustment) and taking appropriate decisions (e.g., AAA, policies and traffic control decisions) as defined by the network operator. Besides, the NASS is used for terminal configuration parameters (dynamic provision of IP addresses) as well as authorization of network access based on user profiles. Moreover, it includes the RACF functions to coordinate the multicast trees creation inside a network. It is important to mention that ACA provides a certain flexibility to use any technique specified by network administrator (e.g., shortest paths techniques [4], [121], flooding-based techniques [231], spanning trees techniques [232], etc.) to create the edge-to-edge multicast trees for data transport across the network. In addition, the RACF defines appropriate resource and admission control system which allows implementing any over-reservation control algorithm such as COR, ECOR or MARA in order to control resource aggregately to scale. It assures the session-to-multicast tree mapping decisions which in turn, are translated into commands and sent to BRs and core nodes for the enforcement which is further detailed in subsequent subsections.

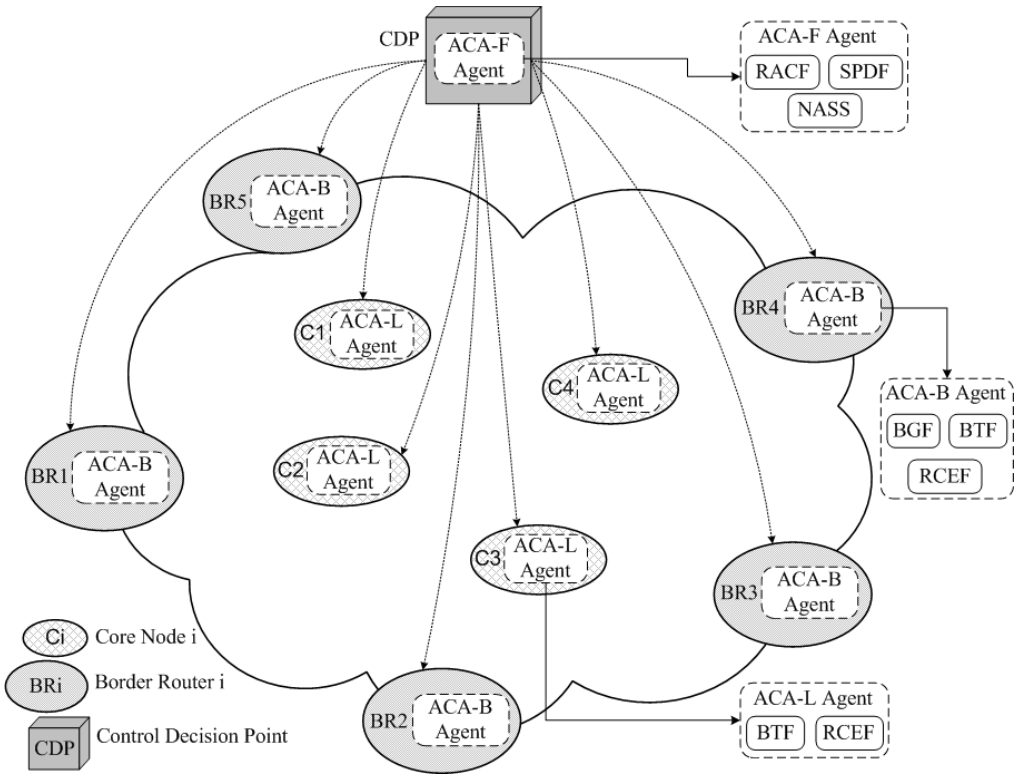


Figure 3.4. Illustration of ACA centralization with TISPAN functional modules mapping.

3.4.1.2 ACA-Light Agent

The ACA-L is a lightweight agent which implements the normal functionalities required in any network normal node (e.g., ingress routers and core routers). These functions include, but are not limited to, control message interception and processing, data forwarding, QoS and multicast trees enforcement, data replication on multicast trees, control feedback, and network element failure notification. Hence, ACA-L implements the RCEF and the BTF functions in a way to properly intercept, interpret and process messages and assure QoS-aware data transport across a network. To assure that control instructions, conveyed through commands in signalling messages, are intercepted on visited nodes along a tree, ACA-L enables UDP port recognition (as routers are permanently listening on UDP port) or the RAO [122] on nodes. Moreover, it is able to interact with RMF [119] to configure schedulers on nodes [64], [65] to assure that each CoS receives the amount of bandwidth allocated to it. Furthermore, it interacts with legacy protocols (e.g., routing protocols) or manipulates appropriate databases on nodes (e.g., MIB, RIB or MRIB, and FIB) according to the control instructions received. It enables nodes to store the ID of previous outgoing interface visited by signalling message, which information is used to avoid asymmetric route issues in reverse path [63], [121]. As an example, ACA-L allows collecting the lists of outgoing interfaces IDs (e.g., RRO [121], [124]) on the trees together with the interfaces' QoS capabilities (e.g., interface capacity) as required by the CDP to build its local NetCIB database.

3.4.1.3 ACA-Border Agent

The ACA-B as a semi-lightweight agent designed for BRs residing at network borders, embeds the ACA-L agent since the latter implements the functionalities which are required at BRs as well. In addition, ACA-B implements the BGF functions in order to execute traffic conditioning (e.g., metering bandwidth usage, reshaping and policing out-of-shape traffics, etc.) to control traffic throughput of every flow according to the control instructions it receives from the ACA-F. Moreover, it bridges inter-domain connectivity on the data plane, knowing that the control plane is managed by the CDP. The inter-domain operations may be based on SLAs/SLSs as we observed in Chapter 2. The interactions between the agents and the overall ACA operations functions are described in the following.

3.4.2 **ACA Operations**

The main objective of this subsection is to describe the ACA operations and the interactions between the ACA-F, ACA-B and ACA-L agents to assure a proper networking mechanism. To facilitate the understanding, our description is illustrated based on Figure 3.5 and uses the ECOR algorithm to demonstrate how to integrate over-reservation scheme to achieve scalable QoS overprovisioning in a network without incurring QoS violation or waste of resources. In this sense, we assume that each BR-Core link inside the network has a capacity *C=1Gbps* and Core-Core has *C=100Mbps*, and implements four CoSs: one Control Signalling CoS (CS) and three service CoSs such as EF, AF and BE. Further, we assume that a fixed amount of bandwidth *b = 1Mbps* is allocated to the control CoS and the weights 40%, 30% and 30% are allocated to EF, AF and BE CoSs, respectively.

As shown in Figure 3.5, ACA follows the traditional master-client control model where the CDP hosting the ACA-F agent is acting as the control decision maker (master), and the BRs and the cores nodes embedding ACA-B and ACA-L respectively, play the role of client for decisions enforcement. The decisions made at the CDP are translated into commands and conveyed to relevant nodes inside the network through appropriate signalling messages. As we further detail in the following subsections, the operations of ACA are divided into Network Initialization phase and Network Running phase.

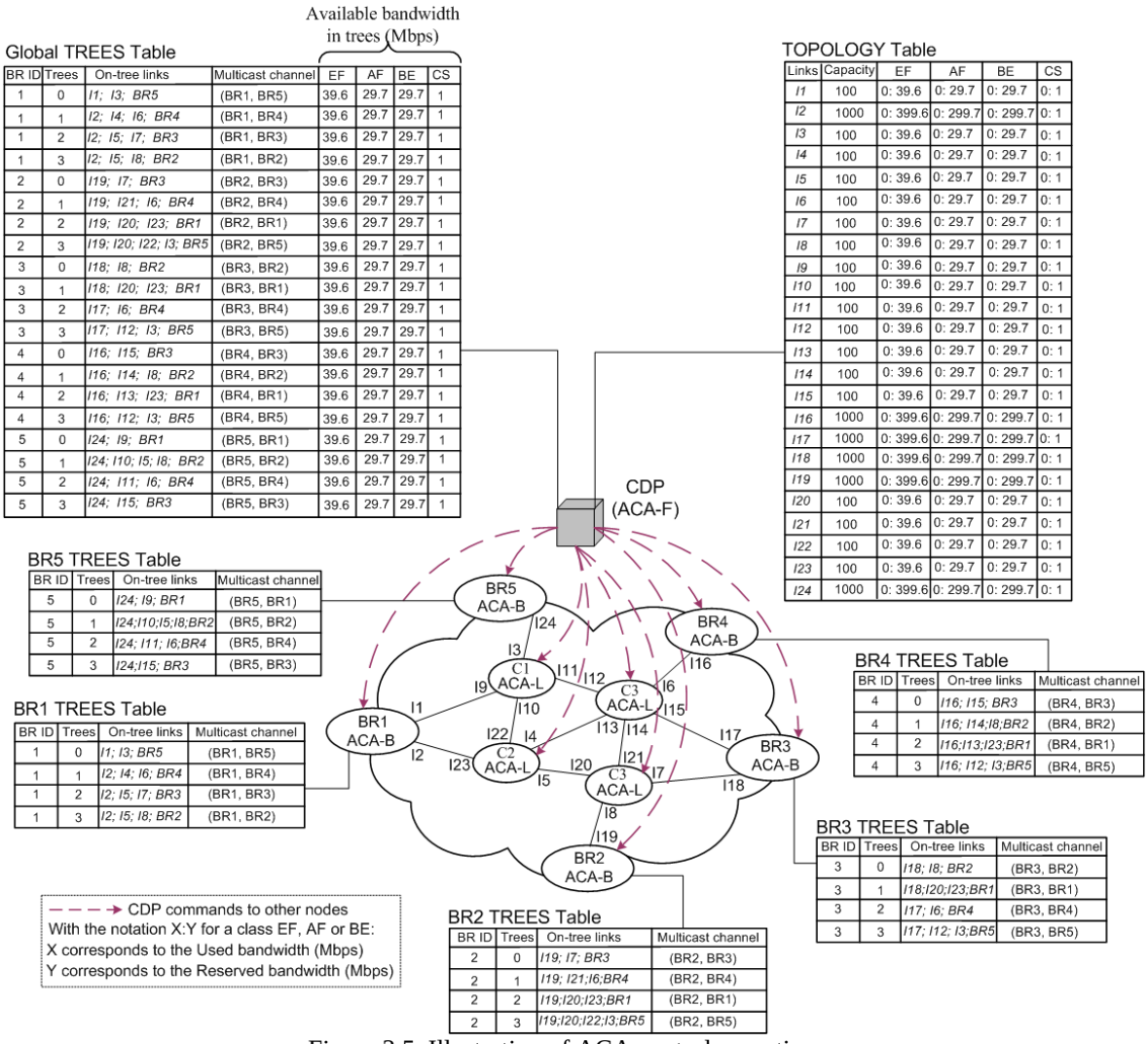
3.4.2.1 ACA Network Initialization Mechanism

The network initialization is characterized by the phase when the nodes inside the network are booting and there is no seesion running yet. Hence, when a node boots up, it announces its presence to the CDP in a message which includes its neighbour nodes. This way, the CDP is aware when all nodes have booted up and uses the information to build the network topology. By using specific

algorithm (e.g., Dijkstra) to the topological information, the CDP can create all the possible edge-edges routes inside the network. In particular, every node implements the ECOR initialization function in equation (3.14) such that the initial reservations are configured on each interface of every node automatically as the node boots up. It is very important to mention that each node is able to modify these configurations upon instructions from the CDP. After all nodes have booted up, the CDP instructs each of the BRs (BR1 through BR5) to create all possible multicast trees from its self to each of the other BRs inside the network as in the following.

To build its trees, a BR sends a packet out on each of its interfaces (except on the inter-domain interfaces) into the network with each packet including the ID and the capacity of the interface through which it was sent. As a packet is travelling across the network, every visited node intercepts it, and forwards a copy on each of its interfaces (except the one on which the packet was received) after appending the ID and the capacity of the interface, leading to the so-called flooding approach. Every node is enabled to drop loop packets when it detects that the received packet already contains an ID of its local interface. This procedure is repeated until the packets reach other BRs. When a BR receives a packet which was not initiated by itself, it sends it to the CDP. This way, the CDP obtains all the edge-to-edge possible routes inside the network together with the corresponding list of outgoing interfaces and the capacities. Then, it uses the interfaces IDs and the corresponding capacities, and applies the equation (3.14) to the capacities to build its initial TOPOLOGY table as in Figure 3.5. Afterwards, it transforms the routes into multicast trees in its global TREES table by assigning a multicast channel and an ID to each of the routes. Note that the CDP can also compute all possible edge-to-edges routes through combination of unbrached edge-to-edge ones as in [162]. Further, the global TREES table records the available bandwidth on each tree as being the bottleneck available bandwidth (minimum available bandwidth) on the tree, obtained based on the information maintained in the TOPOLOGY table and the list of outgoing interfaces on the trees. For simplicity, the databases in Figure 3.5 do not include all the possible trees.

Then, the CDP instructs each BR to enforce the multicast trees inside the network by sending them the assigned channel, the trees IDs and the list of outgoing interfaces. Hence, upon receiving the command, each BR enforces the multicast trees by configuring the MRIB on its local interfaces. After that, it sends the command to the core nodes using source routing to ensure that each multicast channel is properly configured on the correct interfaces as recorded by the CDP. When a tree is successfully created up to a remote BR which must reply, the initiator BR records the information in its local TREES table as in Figure 3.5 and sends the feedback to the CDP for the latter to save the information. This way, the network is initialized and the system run-time operations are described in subsection 3.4.2.2.



Global TREES Table

BR ID	Trees	On-tree links	Multicast channel	EF	AF	BE	CS
1	0	*I1; I3; BR5*	(BR1, BR5)	39.6	29.7	29.7	1
1	1	*I2; I4; I6; BR4*	(BR1, BR4)	39.6	29.7	29.7	1
1	2	*I2; I5; I7; BR3*	(BR1, BR3)	39.6	29.7	29.7	1
1	3	*I2; I5; I8; BR2*	(BR1, BR2)	39.6	29.7	29.7	1
2	0	*I19; I7; BR3*	(BR2, BR3)	39.6	29.7	29.7	1
2	1	*I19; I21; I6; BR4*	(BR2, BR4)	39.6	29.7	29.7	1
2	2	*I19; I20; I23; BR1*	(BR2, BR1)	39.6	29.7	29.7	1
2	3	*I19; I20; I22; I3; BR5*	(BR2, BR5)	39.6	29.7	29.7	1
3	0	*I18; I8; BR2*	(BR3, BR2)	39.6	29.7	29.7	1
3	1	*I18; I20; I23; BR1*	(BR3, BR1)	39.6	29.7	29.7	1
3	2	*I17; I6; BR4*	(BR3, BR4)	39.6	29.7	29.7	1
3	3	*I17; I12; I3; BR5*	(BR3, BR5)	39.6	29.7	29.7	1
4	0	*I16; I15; BR3*	(BR4, BR3)	39.6	29.7	29.7	1
4	1	*I16; I14; I8; BR2*	(BR4, BR2)	39.6	29.7	29.7	1
4	2	*I16; I13; I23; BR1*	(BR4, BR1)	39.6	29.7	29.7	1
4	3	*I16; I12; I3; BR5*	(BR4, BR5)	39.6	29.7	29.7	1
5	0	*I24; I9; BR1*	(BR5, BR1)	39.6	29.7	29.7	1
5	1	*I24; I10; I5; I8; BR2*	(BR5, BR2)	39.6	29.7	29.7	1
5	2	*I24; I11; I6; BR4*	(BR5, BR4)	39.6	29.7	29.7	1
5	3	*I24; I15; BR3*	(BR5, BR3)	39.6	29.7	29.7	1

TOPOLOGY Table

Links	Capacity	EF	AF	BE	CS
I1	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I2	1000	0: 399.6	0: 299.7	0: 299.7	0: 1
I3	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I4	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I5	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I6	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I7	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I8	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I9	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I10	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I11	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I12	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I13	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I14	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I15	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I16	1000	0: 399.6	0: 299.7	0: 299.7	0: 1
I17	1000	0: 399.6	0: 299.7	0: 299.7	0: 1
I18	1000	0: 399.6	0: 299.7	0: 299.7	0: 1
I19	1000	0: 399.6	0: 299.7	0: 299.7	0: 1
I20	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I21	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I22	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I23	100	0: 39.6	0: 29.7	0: 29.7	0: 1
I24	1000	0: 399.6	0: 299.7	0: 299.7	0: 1

BR5 TREES Table

BR ID	Trees	On-tree links	Multicast channel
5	0	*I24; I9; BR1*	(BR5, BR1)
5	1	*I24;I10;I5;I8;BR2*	(BR5, BR2)
5	2	*I24; I11; I6;BR4*	(BR5, BR4)
5	3	*I24;I15; BR3*	(BR5, BR3)

BR4 TREES Table

BR ID	Trees	On-tree links	Multicast channel
4	0	*I16; I15; BR3*	(BR4, BR3)
4	1	*I16; I14;I8;BR2*	(BR4, BR2)
4	2	*I16;I13;I23;BR1*	(BR4, BR1)
4	3	*I16; I12; I3;BR5*	(BR4, BR5)

BR1 TREES Table

BR ID	Trees	On-tree links	Multicast channel
1	0	*I1; I3; BR5*	(BR1, BR5)
1	1	*I2; I4; I6; BR4*	(BR1, BR4)
1	2	*I2; I5; I7; BR3*	(BR1, BR3)
1	3	*I2; I5; I8; BR2*	(BR1, BR2)

BR3 TREES Table

BR ID	Trees	On-tree links	Multicast channel
3	0	*I18; I8; BR2*	(BR3, BR2)
3	1	*I18;I20;I23;BR1*	(BR3, BR1)
3	2	*I17; I6; BR4*	(BR3, BR4)
3	3	*I17; I12; I3;BR5*	(BR3, BR5)

BR2 TREES Table

BR ID	Trees	On-tree links	Multicast channel
2	0	*I19; I7; BR3*	(BR2, BR3)
2	1	*I19; I21;I6;BR4*	(BR2, BR4)
2	2	*I19;I20;I23;BR1*	(BR2, BR1)
2	3	*I19;I20;I22;I3;BR5*	(BR2, BR5)

Figure 3.5. Illustration of ACA control operations.

3.4.2.2 ACA Network Running Mechanism

As the network in Figure 3.5 is initialized and set to operate, all session requests are sent to the CDP to be processed according to the local admission control policies, the requested application QoS requirements (obtained from AF functions via *Gq'* interface -Figure 2.7), the user profiles (obtained from NASS via *e4* interface -Figure 2.7), and a good knowledge of underlying network topology and related links and paths resource status. Hence, all authorized requests are passed to the admission control module RACF embedded in the CDP which retrieves the candidate trees (from its Global PATHS table) that can be used to map the incoming requests according to the available resources on the trees. Hence, when there are sufficient available resources on certain candidate trees, the tree with the highest available resources is selected and the session is thus mapped to the tree. As ECOR is deployed and the resource is over-reserved, the CDP can process several session requests (session setup, release, readjustment) into appropriate trees without QoS reservation signalling as long as the available over-reservation is sufficient in the trees. Thus, ACA

deploys aggregate resource overprovisioning in order to scale by reducing signalling and the related processing overhead. Hence, it is very important to mention that, whenever ACA-F admits, readjusts or terminates a session belonging to a CoS in a given tree, the resource status (e.g., the reserved, the used and the available bandwidth) in each CoS on each outgoing interfaces on the tree, is automatically updated in the local TOPOLOGY table accordingly. Moreover, ACA-F records every active session together with the related flows' information in terms of session's description and QoS requirements (e.g., session ID, flows IDs, session source and destination IDs, the CoS and path mapped to a flow, the bandwidth granted to each flow, etc.). The CDP uses this information to instruct ACA-B agents implementing BGF functions at network border to properly control active sessions (e.g., traffic conditioning – metering, marking, shaping or policing) to assure differentiated QoS granted to session flows during QoS negotiation. Also, the packets that belong to a session are pinned to the tree mapped to the session, assuring that the session enjoys the treatment destined to it in the network. It becomes clear that the trees and interfaces capabilities are accurately maintained in local database in real-time manner without being required to signal the network. Therefore, the CDP is aware of the network accurate resource conditions at any time to prevent wrong admission decisions.

However, in case the CDP's admission functions realize that the available reservation in a requested CoS is insufficient to assure acceptable quality in the candidate trees, the ACA-F agent implementing ECOR through the RACF triggers the ECOR resource over-reservation readjustment functions described in section 3.2.2. The main objective of the CDP is to profit from ECOR to attempt readjustment of the reservation parameters among CoSs on candidate trees to avoid CoS starvation or waste of resource which increase session blocking unnecessarily. This way, ECOR is dynamically invoked to define parameters for reservation readjustment on outgoing interfaces on trees upon need. It is also very important to mention that existing solutions mostly readjust parameters of the outgoing interfaces on a tree by using the parameters obtained based on the bottleneck resource conditions of the tree [125]. We believe that, this limitation is imposed to such solutions, since they mainly rely on the bottleneck information collected through periodic and on-demand measurement techniques (e.g., probing). In contrast, ACA provides a good knowledge of resource statistics on every interface, and uses ECOR to define parameters for an interface on a tree based on the resource conditions of the interface itself.

When new reservation parameters are successfully defined for the outgoing interfaces on a tree, the CDP encapsulates the parameters in appropriate QSPECs objects and associates each QSPEC with the corresponding interface ID. Afterwards, it sends the information to the BR that roots the tree, which first enforces the new configurations destined to its local outgoing interface before forwarding the message down the network to the remaining nodes on the concerned tree. As the

message is travelling along the tree, each visited node, hosting ACA-L agent, retrieves the reservation parameters destined to its local outgoing interface(s) on the tree and enforces the required configurations before forwarding the message. A node at which reservations fail due to node malfunction or link failure must send a failure notification to the CDP. This procedure is repeated on each visited node until the message reaches the corresponding egress BR. Hence, upon receiving the message, the egress BR composes the corresponding response message with a successful reservation flag and sends it to related ingress BR, which in turn, notifies the CDP about the successful reservation operations. Then, the CDP updates the new configuration parameters in its local database accordingly and the request can be accommodated.

3.5 *Performance Evaluation*

The benefits of overprovisioning approach (e.g., COR, ECOR and MARA) and the impact of key parameters that affect its performance were evaluated analytically by comparing COR and ECOR against MARA's algorithm. Moreover, we implement ACA architecture integrating COR, ECOR and MARA to assess and validate results through simulations using the Network Simulator version 2 - ns-2 [233]. Hence, we study performance in terms of control signalling overhead minimization and avoidance of waste of resources, and therefore, the issue of unnecessary increase of session blocking probability.

3.5.1 Assumptions for Analytical Evaluation

As reservation readjustment is usually triggered when the over-reservation is exhausted in a requested CoS on the bottleneck outgoing interface on a desired tree, Figure 3.6 is used to illustrate bottleneck scenarios for our analytical assessment of over-reservation approach. Hence, we assume that a bottleneck outgoing interface I_b is shared by many trees ($T_1, \ldots, T_x, \ldots, T_m$) rooted at various border routers ($BR_1, \ldots, BR_X, \ldots, BR_Z$) which are dynamically injecting traffic into the network. Besides, it is considered that I_b has a capacity C and implements k CoSs as we studied in subsection 3.3.

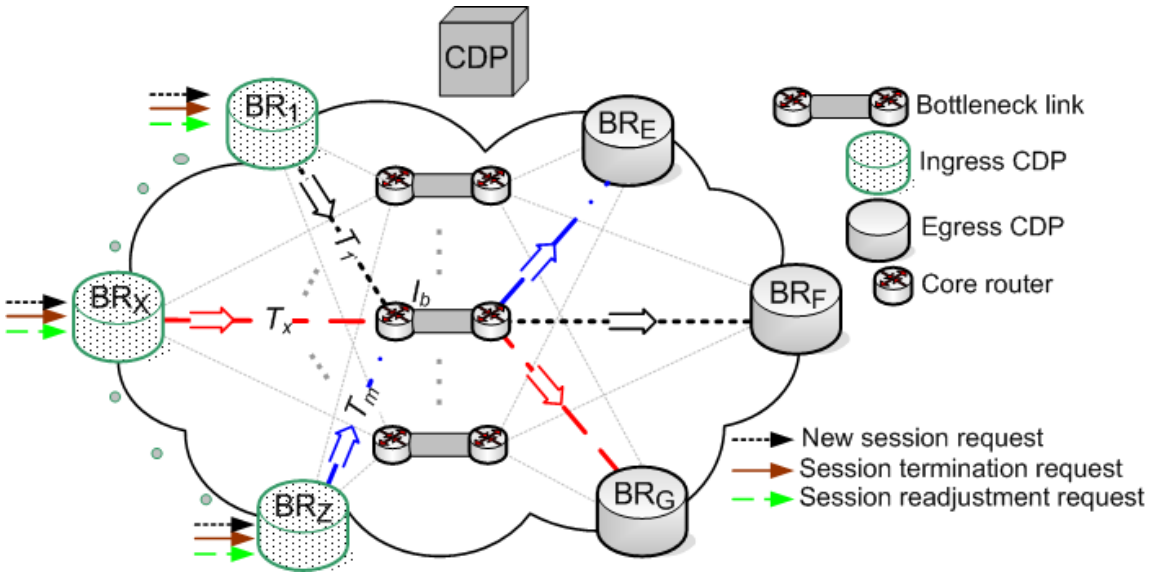


Figure 3.6. Topology for analytical study.

To perform the resources readjustment, the current resource status of each CoS on the bottleneck interface I_b is obtained based on the following assumptions. It is assumed that the current reservation $R_{BW}(j,I_b)$ and the threshold $\chi_{BW}(j,I_b)$ of the congested CoS_j on I_b are:

$$R_{BW}(j,I_b)=U_{BW}(j,I_b) \tag{3.33}$$

$$\chi_{BW}(j,I_b)=U_{BW}(j,I_b) \tag{3.34}$$

In equations (3.33) and (3.34), $U_{BW}(j,I_b)$ is the current used bandwidth in the requested CoS_j and is obtained based on the total amount of used bandwidth $R_{BW}^{Total}(I_b)$ and the sum of the used in the remaining (*k-1*) CoSs on the interface I_b:

$$U_{BW}(j,I_b)=U_{BW}^{Total}(I_b)-(k-1)*U_{BW}(i,I_b) \tag{3.35}$$

where, $U_{BW}^{Total}(I_b)$ is determined by equation (3.21), $U_{BW}(i,I_b)$ is the amount of used bandwidth in each of the remaining CoS_i, $1\le i\le k$, $(i\neq j)$ on the interface I_b and is set based on the total used bandwidth as:

$$U_{BW}(i,I_b)=\left\lfloor \frac{U_{BW}^{Total}(I_b)}{k} \right\rfloor \tag{3.36}$$

Considering that the requested CoS_j is currently set with no available resource as in equations (3.33) and (3.34), the current peak threshold $\chi_{BW}(i,I_b)$ of each CoS_i of the remaining (*k-1*) CoSs on the interface is assumed to be:

$$\chi_{BW}(i,I_b) = U_{BW}(i,I_b) + \frac{\Delta_T(I_b)}{k-1} \tag{3.37}$$

Thus, in ECOR, the residual over-reserved bandwidth $Residual\_R_{BW}(i,I_b)$ in each of the remaining CoS_i is obtained based on the unused resource in equation (3.37) and is configured as:

$$Residual\_R_{BW}(i,I_b) = \frac{\Delta_T(I_b)}{k-1} \tag{3.38}$$

In COR or MARA, the residual over-reserved bandwidth $Residual\_R_{BW}(i,I_b)$ in each of the remaining CoS_i is considered as half of the latest over-reservation computed by the algorithm, that is, half of the previous output of the equation (3.5). This way, the residual over-reserved bandwidth and the used bandwidth in each of the remaining CoS_i are known, and the current reservation $R_{BW}(i,I_b)$ of each of the remaining CoS_i in ECOR, COR or MARA is configured as:

$$R_{BW}(i,I_b) = U_{BW}(i,I_b) + Residual\_R_{BW}(i,I_b) \tag{3.39}$$

Based on these assumptions each algorithm computes new surplus of bandwidth to readjust the reservation parameters of the CoSs on the interface. Recall that, under *Low resource utilization* conditions, ECOR computes new surplus of reservation using the equation (3.24). COR uses equations (3.26) and (3.27), while MARA uses the equations (3.25), (3.26). Under *High Utilization* conditions, ECOR computes new surplus using the equation (3.29), COR uses equation (3.32) and MARA uses the same functions as in low utilization conditions.

3.5.2 **Analytical Results**

Based on the configurations in Table 3.1 and using the scenario of Figure 3.6, Figure 3.7 plots the probability of QoS reservation events occurrence as a function of network bottleneck interface's resource utilization level on communication paths.

Table 3.1. Available resource scenario configuration parameters.

$k = 3$	Number of service CoSs implemented on the interface.
$\bar{r} = 1$	Mean bandwidth requested by each session (Mbps).
$\mu = 1/4$	Mean service rate per session (requests/time unit).
$\lambda_i = 20$	Session requests arrival rate to a CoS_i (requests/time unit).
$C = 1000$	Interface capacity (Mbps).

As one can see in Figure 3.7, the more there are unused session slots on the bottleneck interface, the lower the probability of signalling occurrence is, regardless of the algorithm in use. This shows that, the more resources are available, the more each algorithm over-reserves. However,

we observe that ECOR outperforms both COR and MARA in terms of probability of signalling events occurrence. This is due to the fact that ECOR allows for over-reserving as much resources as a CoS requires, while it is able to efficiently reuse the residual resources among existing CoSs dynamically to prevent waste of resources or CoS starvation, as we explained earlier. Both COR and MARA over-reserve a relatively small portion of unused resources each time in order to reduce the impact of CoS starvation and waste of resources. The main reason behind this behavior of COR and MARA is that efficient over-reservation scheme strongly requires resource utilization statistics in each CoS on each bottleneck interface inside a network on *real-time* basis and *without signalling* the network. However, this requires a new approach for resources information transfer between network elements, that will be detailed in Chapter 4.

We observe that MARA is subject to high probability of signalling when Q is smaller than 400, i.e. when the network is close to congestion. This is due to the fact that MARA computational procedure, consisting of taking a certain amount of resources from the other CoSs to decongest a given CoS, shows serious inefficiency problems when the network is close to congestion. This inefficiency induces waste of resources and CoS starvation problems as we will see in Figure 3.10. Indeed, MARA shows different behaviors when Q is higher than 400, lower network utilization phase. It is important to note that MARA seems to outperform COR when Q is higher than 400, but it is not. We must therefore recall that, for simplicity in this analytical model, a particular CoS_i always congests first to trigger the control events on the bottleneck interface. This obviously favors MARA which removes a certain amount of resource from each of the remaining CoSs to increase the threshold of the always congested CoS_i. In other words, the scenario allows MARA to increase the threshold of the concerned CoS_i much more than the COR, which seeks more balanced control by using weights of CoSs; therefore MARA, can over-reserve more than the COR since both use the same function in equation (3.5) to compute surplus of reservation. In real network scenarios, it is less likely that only a particular CoS gets congested all the time on a long term. Therefore, a steadier behavior of COR, ECOR and MARA will be further observed through large scale simulation results using the ns-2 in Chapter 4, since the main objective of this chapter is to provide insight of major issues that can be addressed to improve scalability in current and future network scenarios. This also explains that with higher interface capacity, the more suitable over-reservation approach will apply. More importantly, QoS control would not be necessary if network resource were unlimited. One can also notice that the ECOR reservations are “Not a Number - NaN” when the number of unused slots is 571 or 666 (too high), which implies that resource control would not be necessary if link bandwidth were unlimited.

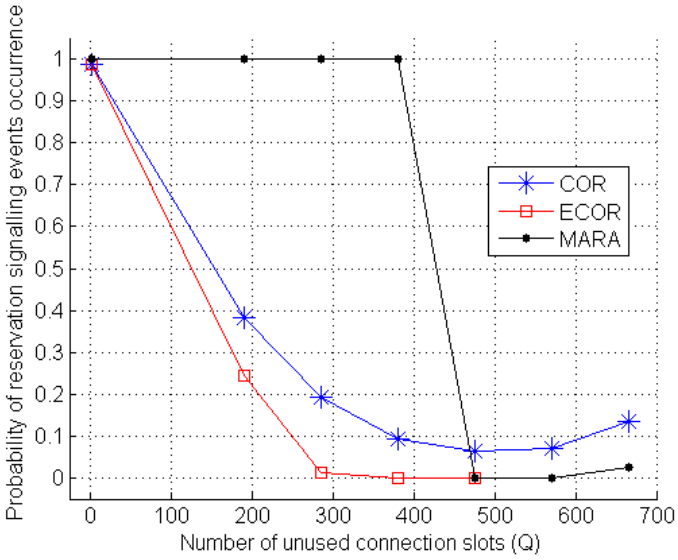


Figure 3.7. Effect of resource utilization level on reservation signalling frequency.

Figure 3.8 analyses the effect of session lifetime (e.g., short-lived, long-lived, etc…) and the suitability of resource over-reservation in dynamic network scenarios. The model is configured with the parameters in Table 3.2.

Table 3.2. Lifetime scenario configuration parameters.

$k = 8$	Number of CoSs implemented on the interface.
$\bar{r} = 1$	Mean bandwidth requested by each session (Mbps).
$\lambda_i = 70$	Session requests arrival rate to a CoS_i (requests/time unit).
C = 1000	Interface capacity (Mbps).
Q = 666	Interface utilization level.

We observe in Figure 3.8 that, in a scenario where most of sessions are short-lived (the higher the service rate, the shorter the lifetime), the probability of signalling occurrence is lower than when sessions' lifetime increases. This shows that short-lived sessions leave the reservations more quickly, and these reservations can be reused for accommodating other incoming requests without signalling the network.

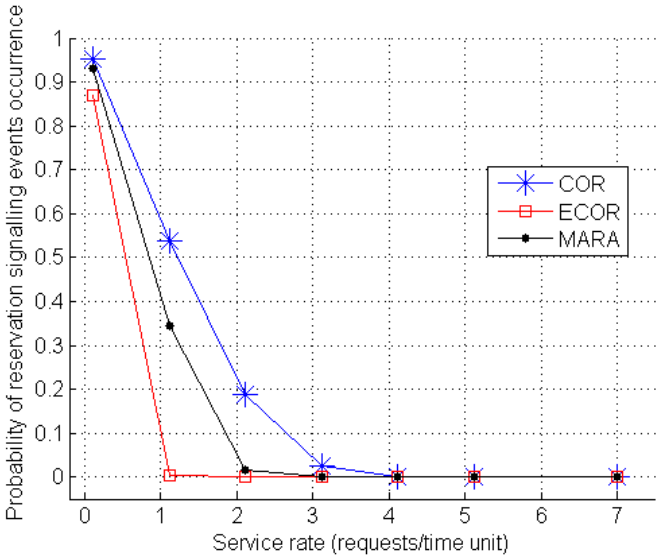


Figure 3.8. Effect of sessions lifetime on reservation signalling frequency.

Figure 3.9 analyses the impact of bandwidth demand of the services requested over the network. In this sense, one can see that the probability of QoS signalling events occurrence increases with the increase of the demand. This explains the fact that the over-reserved resource are consumed more rapidly, and hence, it justifies the increasing need for reservation parameters readjustment among the compteting CoSs.

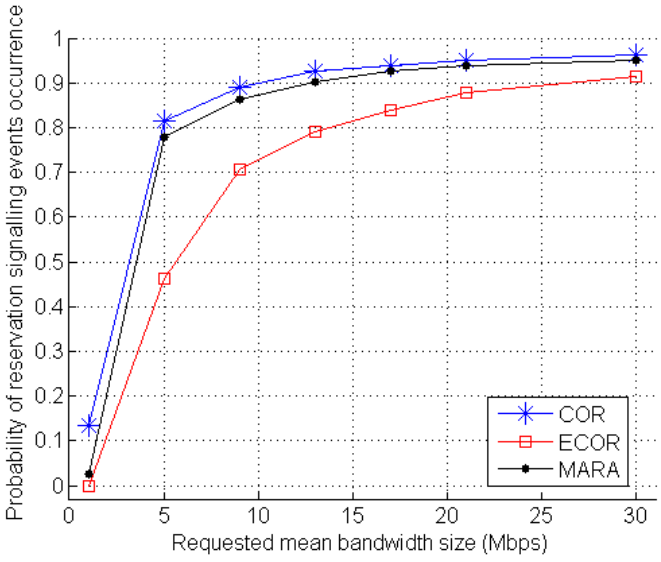


Figure 3.9. Effect of bandwidth demands on reservation signalling frequency.

Figure 3.10 plots the number of session requests that may be blocked unnecessarily when the requested bandwidth by incoming services is available on the bottleneck interface of a desired path. This is the major issue of CoS starvation, waste of resource or unnecessary increase of blocking probability. One can see that neither COR nor ECOR blocks any request when there is a free slot, knowing that each request only demands one slot. This confirms our study in sections (3.1) and

(3.2): COR and ECOR are able to use the sum of all residual resources from all existing CoSs in an efficient manner. In contrast, when the number of unused slots is smaller than 400 (note that each slot has 1Mbps), MARA denied all the requests unnecessarily (please see the segment of line ($y = x$) where y is the number of blocked while x slots were free). This is due to the fact that MARA is not able to collect all residual resources from existing CoSs, which is crucial when links are close to congestion. Indeed, as the number of free slots increases beyond 400 and there is a larger amount of available resources on the bottleneck interface, MARA starts admitting requests since it can take enough resources from the other existing CoSs.

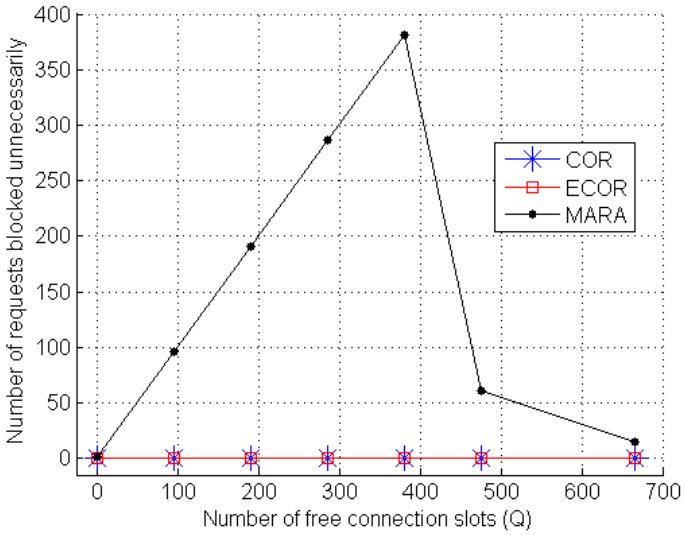


Figure 3.10. Unnecessary increase of requests blocking or waste of resources.

3.5.3 **Simulation Scenario**

In order to show stable results in dynamic network scenarios, the ACA architecture and the over-reservation algorithms (COR, ECOR and MARA) were developed in the ns-2 [233]. The simulations were carried out using 4 randomly generated topologies (number of ingress routers ranging from 3 to 6; core routers: 5 to 15, egress routers: 3 to 6, and one CDP per network to take overall control). One of the simulated network topology is presented in Figure 3.11.

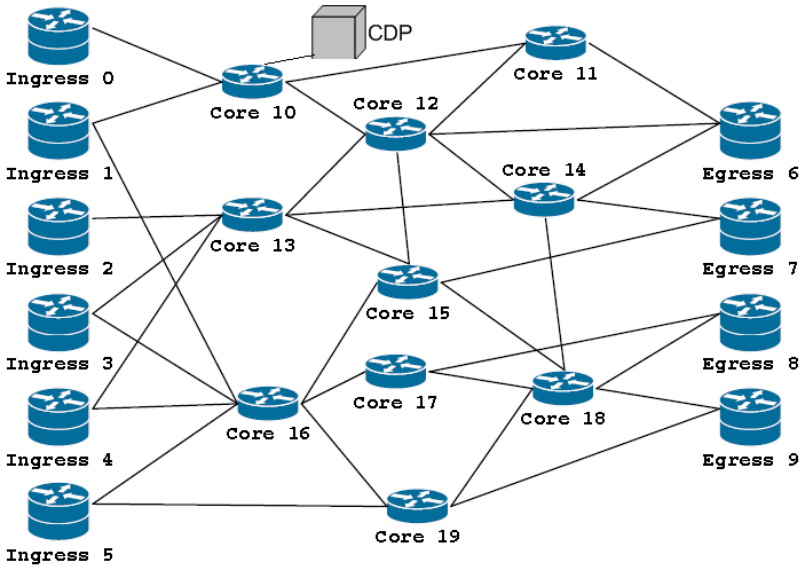


Figure 3.11. Example of ACA simulation network topology.

For simplicity, 4 CoSs configurations are implemented in each network interface, as in the following: one CS, one EF, one AF and one BE [234] under the WFQ scheduling discipline [64]. Considering that bottleneck interfaces' locations inside real networks depend on traffic load dynamics inside the network, each network interface is configured with the same capacity *C=1Gbps*; three different traffic types such as Constant Bit Rate (CBR), Pareto and Exponential are randomly generated based on Poisson processes. Traffic requests belonging to various CoSs, are generated using uniform distribution between 128Kbps and 8Mbps, and are mapped to ingress-egress pairs based on Poisson processes. Among the requests generated, 30% are long-lived sessions (with lifetime of the whole simulation time), 40% are relatively long-lived sessions (with lifetime of 60 minutes) and 30% are short-lived sessions (with lifetime of 10 minutes). It is very important to note that the requests to a network are sent to the corresponding CDP, since the overall control is centralized on the CDP.

We obtain network overall resource utilization in percentage (%) in each simulation results, it is computed as a mean of the resource utilization level on the bottleneck interfaces of all trees inside the network. The studied metrics include the QoS reservation signalling overhead (signalling events and load), the overhead reduction by ECOR in relation to both the COR and MARA, and the issue of waste of resources shown in terms of the number of sessions blocked unnecessary. To show more accurate results, each simulation runs 10 times with different seeds of random mapping of requests to CoSs, and ingress egress pairs for each topology. Then, the mean values are plotted for all topologies and seeds with a confidence interval of 95%.

3.5.4 **Simulation Results**

Figure 3.12 plots the number of QoS reservation signalling events at different network resource

overall utilization levels, and Figure 3.13 shows the corresponding signalling message load. Hence, we observe that, below 60% of network resource utilization level, COR and MARA generate a similar number of reservation signalling events and load. This is mainly due to the fact that COR and MARA use the same function (equation (3.5)) to define resource surplus for CoSs. This shows that the main benefit of COR algorithm over MARA is not on the minimization of the signalling overhead. Rather, it resides in the way that COR distributes residual resources among CoSs to prevent waste of resources or unnecessary increase of session blocking as depicted in Figure 3.16. Hence, one can see that COR triggers more reservation readjustment signalling messages than MARA in the situations of high network resource utilization situation (above 60%). It is very important to notice that this does not mean that MARA outperforms COR in terms of signalling overhead. MARA places less signalling events since it fails to efficienly redistributing residual reservations among CoSs, especially when network is close to congestion or get congested, which is a strong limitation since it leads to a significant waste of resources as in Figure 3.16. ECOR does not issue any reservation signalling messages up to about 85% of network overall resource utilization level. This is because ECOR over-reserves as much resources as possible to each CoS at system initialization, in contrast to COR and MARA. The data points are not visible in these cases due to the log scale plotting of zero (0). Moreover, ECOR maintains its superiority over COR and MARA in terms of signalling events and load overhead under high network resource utilization level. Thus, ECOR proves that one should be able to over-reserve as much resources as possible to effectively reduce the QoS signalling and related processing overhead.

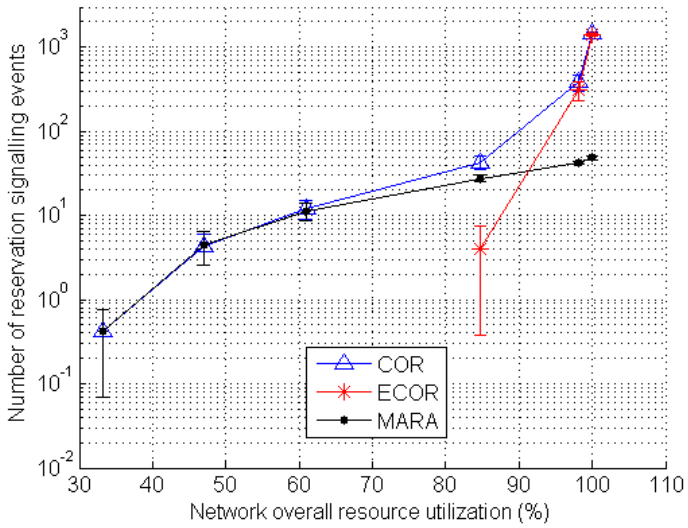


Figure 3.12. Number of reservation signalling events.

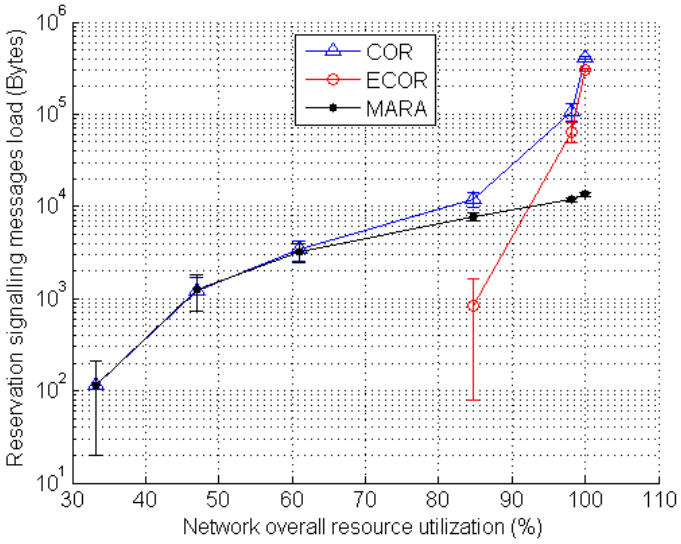


Figure 3.13. Reservation signalling load.

In order to better observe the performance in terms of signalling overhead reduction by ECOR against COR or MARA, Figure 3.14 plots the percentage of signalling events reduction of ECOR over COR or MARA, while Figure 3.15 plots the corresponding signalling load reduction. We observe that ECOR is able to reduce the QoS reservation signalling events of COR and that of MARA between 3% and 100%, depending on the level of network resource overall utilization. While the signalling events number is reduced between 2.9% and 100%, the corresponding signalling load is reduced between 11% and 100%. It is therefore very important to notice that ECOR reduces even more in terms of signalling load as we expected, since ECOR deploys less reservation parameters (e.g., ECOR has no threshold parameter per CoS) than COR and MARA, and thus, the size of the signalling messages generated by ECOR is smaller. It is also important to mention that the data points of MARA around 85%, 97.5% and 99.98% are not visible due to log scale plotting of negative values, since ECOR generates more signalling events than MARA under the situations of high resource utilization, when MARA readjusts less the reservations and leads to CoS starvations and unnecessarily increase of session blocking probability.

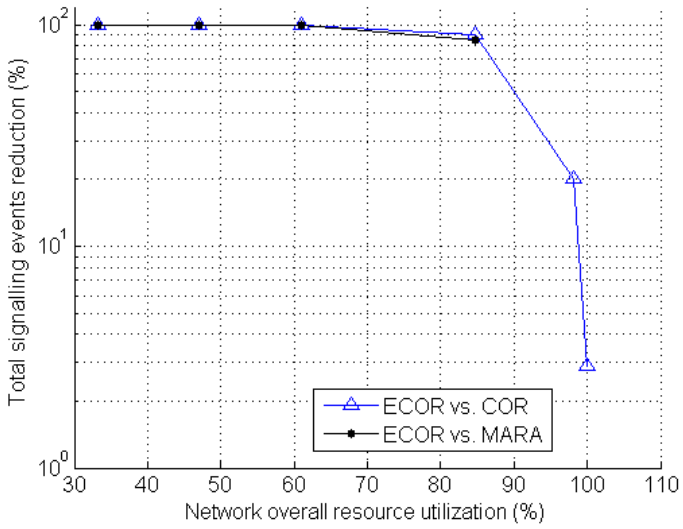


Figure 3.14 Reduction of signalling events number of ECOR vs. COR and MARA.

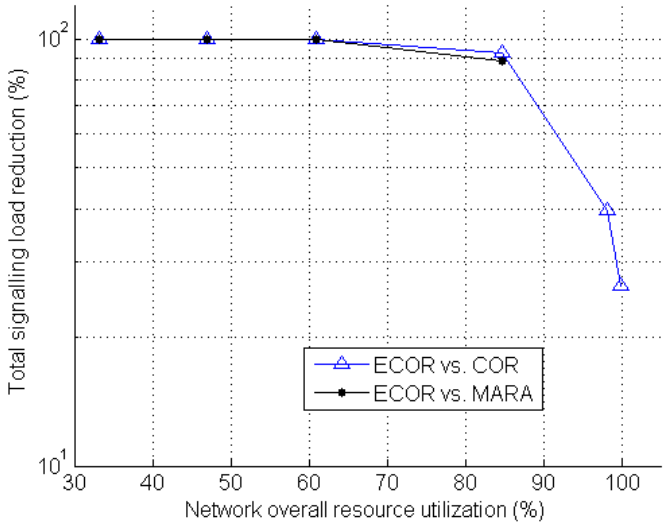


Figure 3.15. Reduction of signalling load of ECOR vs. COR and MARA.

Moreover, Figure 3.16 plots the number of service requests that were blocked when the requested resources were available in the network. As we detailed in subsections 3.1, 3.2 and 3.3, COR and ECOR are able to efficiently reuse residual reservations while MARA fails to do so. As a consequence, MARA increases service blocking probability unnecessarily when network is close to congestion or congested.

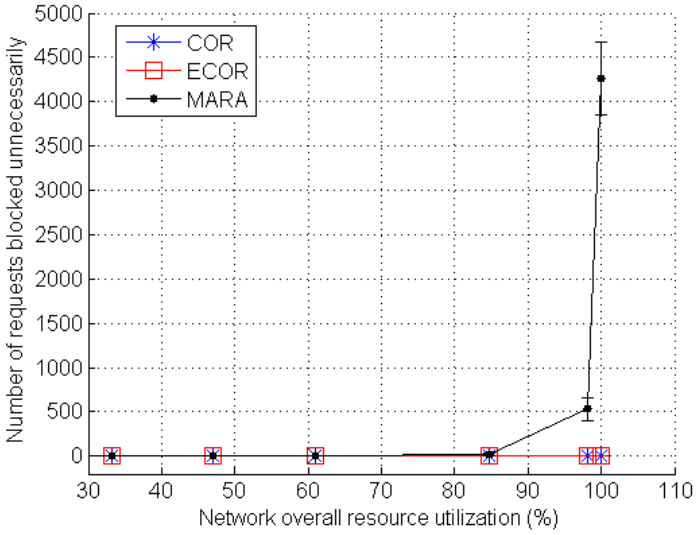


Figure 3.16. Number of session requests blocked unnecessarily.

Hence, it becomes clear that ACA integrating ECOR is able to allow for optimizing network control performance.

3.6 *Conclusion*

In this chapter, we showed that efficient aggregate resource over-reservation algorithms can effectively prevent CoS starvation, waste of resources or unnecessary increase of service blocking probability, while the QoS reservation signalling and the related overhead minimization can be optimized. In particular, we proposed COR which properly deals with CoS starvation and waste of bandwidth, and compared its results with MARA. However, COR and MARA prevent from over-reserving too much resources to CoSs, and thus, fail to allow for optimizing the performance in terms of signalling overhead minimization. Therefore, we proposed the ECOR which enables for reserving to each CoS as much resources as possible, and efficiently redistributes the reservations among CoSs in a way to avoid CoS starvation and waste of resources. ECOR demonstrates its capabilities in allowing for optimizing the overall QoS reservation performance through analytical and simulation results.

Moreover, we studied that over-reservation approaches strongly require network resources statistics in each CoS on each relevant network link in real-time manner. In this sense, we propose the ACA centralized architecture in which a central CDP is enabled to create and maintain many multicast trees inside a network under its control. Thus, packets that belong to a session mapped to a tree are forced to follow that tree. Moreover, it records in its NetCIB database the lists of outgoing interfaces that lie on the existing trees, and automatically updates the resources statistics on every outgoing interface of a tree as soon as it processes a session on the tree. This way, ACA

maintains a good knowledge of network topology and the related links resources statistics in real-time manner without undesired signalling overhead inside the network. This support for over-reservation motivated our implementation of COR, ECOR and MARA in ACA to show results as a use case. Thus, ACA integrating ECOR demonstrated a promising solution for optimization in terms of signalling overhead reduction without wasting resources or QoS violations. Nonetheless, centralized systems such as ACA suffer from single point of failure issues while a central controller can be easily bottlenecked in large network scenarios. Therefore, further investigations to decentralize network control, while assuring efficient support for aggregate over-reservation functions, remained our major concerns in the rest of the Thesis.

Chapter 4

A Self-Organizing Multiple Edge Nodes Mechanism

As we studied in Chapter 1 and Chapter 2, decentralization of network control is promising for the current and future network scenarios since it can scale better than the centralized paradigm. However, decentralization imposes many challenges. On one hand, while it distributes the control load to improve scalability, it requires a correct synchronization of control data between distributed network CDPs to avoid wrong and incompatible decisions. Hence, decentralization must be carefully designed to prevent excessive synchronization signalling and the related processing overhead to effectively scale. On the other hand, resource over-reservation approach consists of reserving more resources than a CoS may currently need, and thus, allows for reducing QoS reservation signalling overhead. However, the approach strongly requires a good knowledge of network topology and the related links resources statistics in each CoS, on *real-time* basis without signalling the network. Therefore, a new decentralized resource control approach is required to address these issues.

Moreover, we saw in Chapter 2 that, NSIS protocol suite [119] provides flexible and extensible support for QoS signalling services over heterogeneous QoS Models (QoSMs) through the *QoS Signalling Layer Protocol* (QoS-NSLP) [123]. Examples of QoSMs include the IntServ, the DiffServ, the ACA in Chapter 3, and the *Self-Organizing Multiple Edge Nodes Mechanism* to be described in this chapter. Basically, the QoS-NSLP signalling protocol allows for carrying control information specific to a QoSM using appropriate QSPEC objects which are interpretable by the RMF [119] implemented in that domain. This way, NSIS is flexible and extensible, such that, new

control parameters and signalling message processing rules can be defined to enhance control performance inside a network without requiring changes in the neighbouring networks.

This chapter introduces the ACOR, a generic-purpose mechanism to provide communication and synchronization between the edge nodes in a network to support decentralized resources control. The main idea consists in enabling multiple distributed CDPs (e.g., at network edges) to selectively cooperate and self-control, by jointly exploiting control data inside a network in a way that allows each CDP for maintaining a good view of network topology and the related links' resources statistics in each CoS, on real-time manner, with low signalling overhead. This way, ACOR aims to provide good support for network resources key control sub-systems (e.g., traffic engineering, QoS over-reservation, end-to-end transport control, link capacity planning, etc.), such that system overall performance can be improved in a flexible and cost-effective manner. Hence, ACOR implements the ECOR algorithm introduced in Chapter 3 to minimize QoS reservation signalling overhead without incurring QoS violation, CoS starvation and waste of resources. Moreover, we propose a concept of virtual resource sharing, the VOPR, which allows for keeping low rate of synchronization signalling between CDPs peers.

In order to demonstrate the effective support of ACOR in terms of resource over-reservation integration besides the ECOR implementation, we also implement the over-reservation algorithms proposed in Chapter 3, the COR [235] and MARA [125]. Furthermore, we introduce the ACOR-P, an NSIS compliant signalling protocol, which is designed to support the overall control mechanism proposed in this Thesis. The evaluation of ACOR was carried out through analytical and simulation studies, which analyzed the decrease in the control signalling and the improved resource utilization without damaging system performance in terms of waste of resources, unnecessary blocking and QoS violation.

This chapter is organized as follows. We describe the ACOR control mechanism integrating the ECOR algorithm in section 4.1, and present the ACOR-P protocol in section 4.2. Then, the section 4.3 provides an analytical model of ACOR and section 4.4 addresses the performance evaluation with both the analytical and simulation results. Finally, section 4.5 concludes the chapter.

4.1 *ACOR Control Mechanism*

The ACOR specifies a decentralization control mechanism which deployes differentiated QoS provisioning in class-based networks by dynamically controlling aggregate bandwidth over-reservation, seeking significant reduction of control overhead. The intelligence in ACOR is pushed to the network border, where each node (i.e., ingress/egress node) hosts the so-called CDP entity

and interior/core nodes are left simpler. The CDPs are responsible for making policies and control decisions, which in turn, are translated into commands and conveyed in signalling messages to the core nodes for the enforcement. As we illustrate using Figure 4.1 to facilitate the understanding, all available CDPs cooperate as a means to dynamically exchange appropriate control information for synchronization to changes of network resource states, and therefore, to assist control decisions with accurate information in distributed manner. It is very important to mention that every CDP in ACOR creates and manages multiple edge-to-edges multicast trees which are used, not only for group communication purposes, but more primarily to force every packet of a session mapped to a tree to follow the desired tree. This way, each CDP is able to use the correlation patterns of the trees and the traffic dynamics in each correlated tree to infer resource statistics in each CoS on each interface inside the network, thus providing a good view of network topological and related resource information. A correlation pattern of trees on an outgoing interface reflects the number of trees that share the interface, the interface's *Sharing Factor*, together with the trees' relevant information, such as the trees IDs and the CDPs from which the trees originate.

ACOR is able to achieve this with significantly low signalling overhead mainly through a two-layering control approach, which bases on the concept of aggregate resources control in class-based networks. On one hand, ACOR implements ECOR techniques for dynamic control of aggregate bandwidth over-reservation, and thus, allows for minimizing the rates of QoS reservation signalling in the sense to optimize performance (e.g., with low CPU and memory consumption by networking processing), without QoS violation or waste of resources. On the other hand, it introduces the VOPR concept [236], which consists in a way of virtually allocating a share of aggregate over-reservations of every CoS on each outgoing interface to each edge-to-edges tree that uses the interface. This way, a CDP can process several session requests to a CoS on a tree without requiring synchronization as long as the VOPR of the tree for the CoS is not exhausted, and thus allows for reducing the synchronization frequency. Moreover, the ACOR synchronization operation is selective, that is, only the CDPs, which are correlated with the information to be updated, are dynamically included in a collaboration group such that the information is not broadcasted unnecessarily.

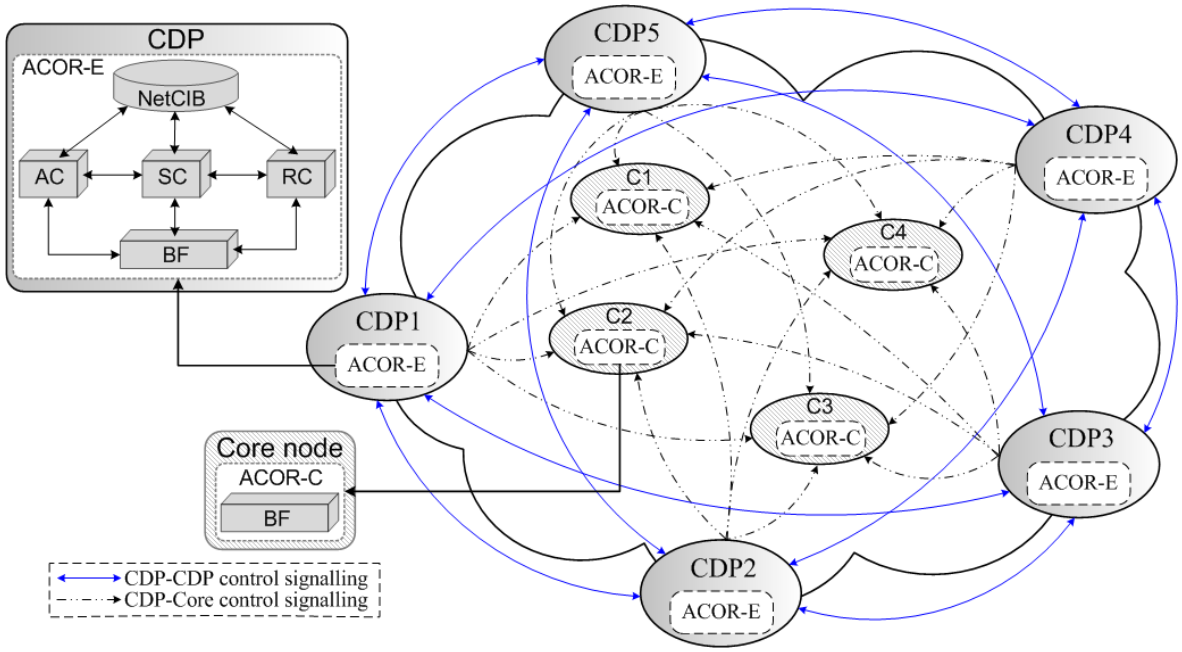


Figure 4.1. Illustration of ACOR decentralized network topology.

As in Figure 4.1, the ACOR architecture is composed of the following components and control functionalities:

- *Network Control Information Base* used by each CDP to store and maintain appropriate control information inside the network. The NetCIB of a CDP is mainly composed by four information tables, including: 1) **TREES table** to store the CDP's selected trees and the related control information, where each tree is associated with a list of correlated CDPs which is dynamically used to assist the selective cooperation between CDPs (two CDPs are considered correlated when their selected trees happen to correlate by sharing outgoing interface(s)); 2) **TOPOLOGY table** to store the IDs of the outgoing interfaces of the CDP's selected trees and the related control information (e.g., interface capacity, reserved and used bandwidth, etc.); 3) **VOPRS table** to store the VOPR of each selected tree for each CoS on each outgoing interface on the tree; and 4) **SESSIONS table** to store the characteristics and QoS requirements of flows composing admitted sessions.
- *Admission Control (AC)* to accept or deny a service request depending on the service requirements and network resource availability.
- *Synchronization Control (SC)* for the cooperation between CDPs to assure a proper synchronization of topology and the related links' resource status through the concept of VOPR.
- *Resource Control (RC)* responsible for the QoS over-reservation decisions and the dynamic readjustment of the reservations upon need.

- *Basic Functions (BF)* such as packet forwarding, QoS reservations and multicast trees enforcement, failure events reports, and interaction with legacy control functions (e.g., packet schedulers, buffer management, etc.).

These features are implemented in software agents classified as ACOR-Edge (ACOR-E) and ACOR-Core (ACOR-C). The ACOR-E is a statefull agent embedded in each ingress and egress node (i.e., CDP), and therefore, implements all the components such as NetCIB, AC, SC, RC and the BF components as depicted in Figure 4.1. The ACOR-C is a lightweight-state agent embedded in all core nodes and implements the BF component, seeking to react upon ACOR-E decisions enforcement requests, to assure packets forwarding, operations feedback (successful/unsuccessful) or failure reports. As in Figure 4.1, the focus of this Thesis is on a single control domain which is not limited to Autonomous System, but can be defined by the network administrator (e.g., a cluster or an area as in OSPF-based domains [4]). This assumes that each domain can deploy its own control model, and inter-domain connections can be assured in many different ways (e.g., using SLAs/SLSs) depending on specific design objectives and control policies. In this sense, we use Figure 4.2 to show a larger ACOR-enabled network spanning three different control domains. Each domain deploys multiple border nodes which embed the ACOR-E, while the core nodes implement the ACOR-C agents. Moreover, the domains inter-connect through redundant links, which is important to improve robustness and availability in networks.

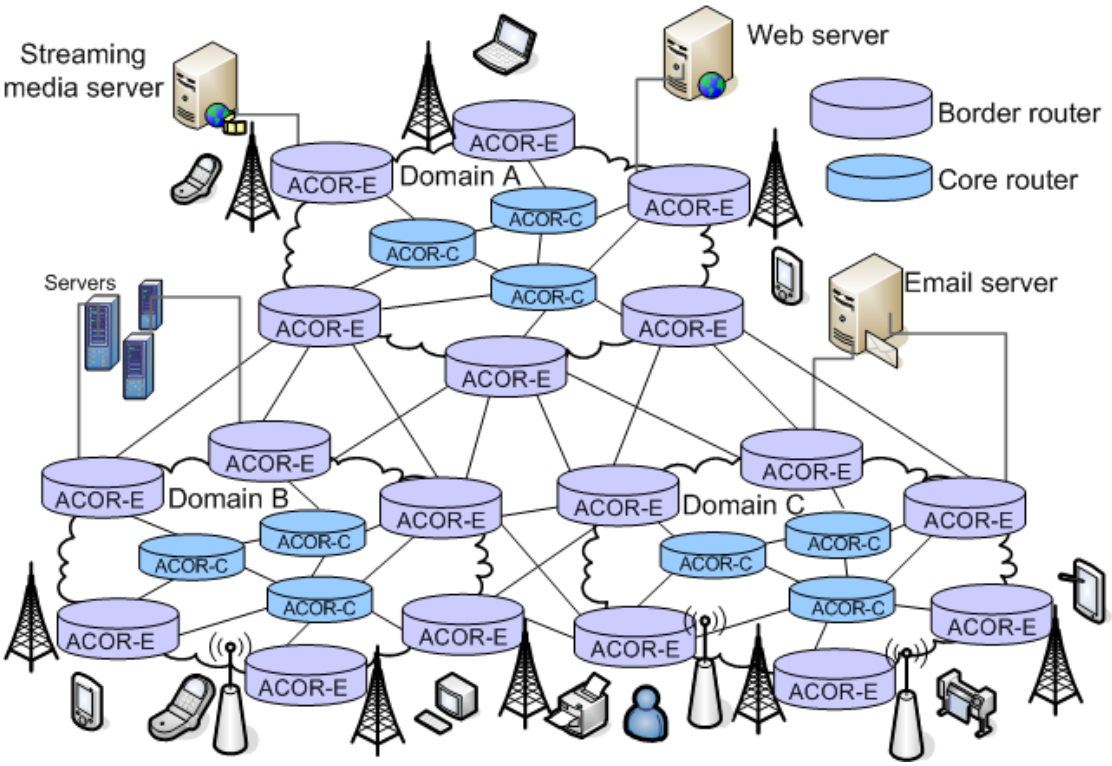


Figure 4.2. Illustration of large ACOR enabled network scenario.

The ACOR operations are divided into network initialization and network running phases and are described in the following.

4.1.1 ACOR Operations at Network Initialization Phase

In order to ease the understanding, our description is illustrated using Figure 4.3. We consider that each interface in the network has a capacity (*C=1Gpbs*), and implements one dedicated Control Class for control packets and k service CoSs (we use 3 classes as example, such as one EF, one AF and one BE CoS [234]). Considering the integration of ECOR for the resource control, we allocate a fixed amount of bandwidth *(b=1Mbps)* to the control CoS and assign a weight 40%, 30% and 30% to the EF, AF and BE CoSs, respectively. The term CoS will be used to refer to a service CoS, unless if it is indicated as a control CoS or CS. The ACOR initialization functions are divided into two parts: *1) Initial Resource Control and Basic Functions* which assure the bandwidth-aware multicast trees creation and the selection of the best trees as in *step (a)* in Figure 4.3; *2) Initial Synchronization Control Functions* by which the CDPs exchange initial control data as in *step (b)*, and build initial knowledge of the network and related resources, as being, the NetCIBs creation in *step (b)*. Further details are provided in the following.

4.1.1.1 Initial Resource Control and Basic Functions

To assure a dynamic initialization and adaptation of the network as nodes (e.g., CDPs or core routers) boot up, each node publishes its presence with its connection information (e.g., list of neighboring nodes) using a flooding mechanism. Flooding approach is preferred for a fast and reliable notification of presence to all CDPs inside a network (similarly to the OSPF [4]). This way, each CDP is able to realize when a node boots up or when all nodes have booted up, so as to maintain a consistent view of the network.

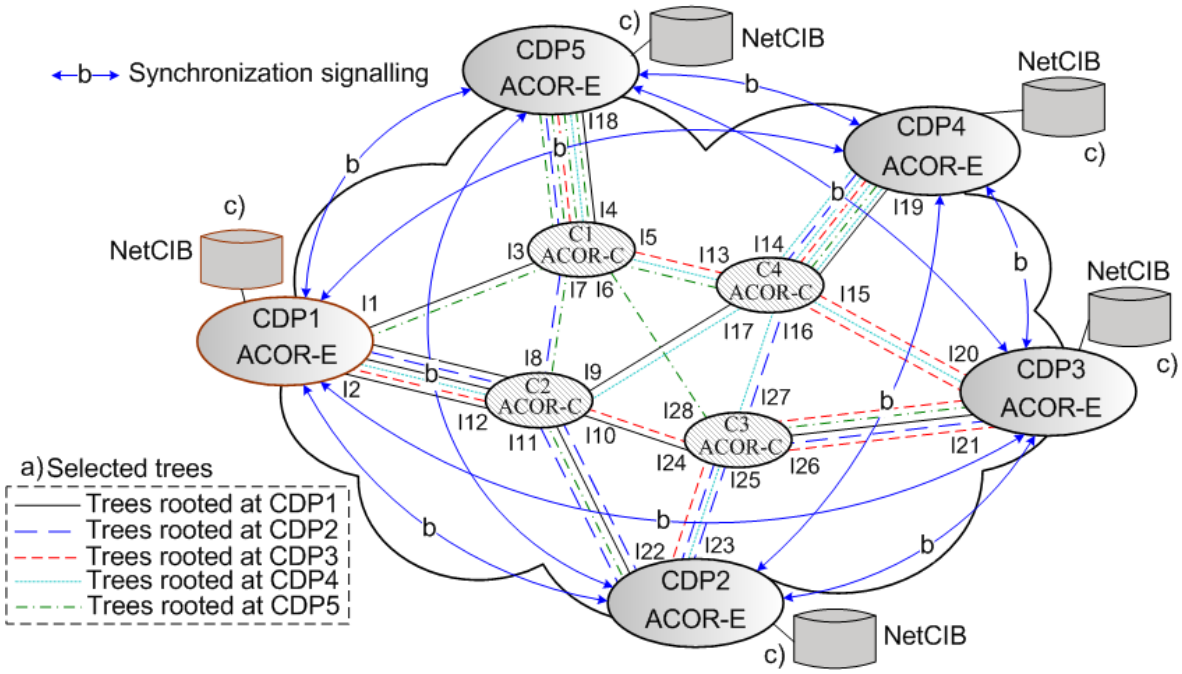


Figure 4.3. Illustration for ACOR operations.

After all nodes have booted up, the RC module exploits the ACOR-P signalling protocol described in section 4.2, and discovers all possible edge-to-edge routes from itself to each CDP inside the network using a flooding-based technique similar to the approach in [236]. In particular, it creates a signalling message and sends it down the network through each of its interfaces except the inter-domain interfaces. The information carried in the message includes the weights assigned to each CoS, the amount of bandwidth *b* dedicated to the control CoS, the ID and the capacity of the interface through which the message is sent. Notice that the interface ID collected must be stable and unique during the whole network operations for control stability, and may be the IP address configured on the interface, the interface Medium Access Control (MAC) address or the ID of the node. Hence, when visited by a message, any node implementing the BF functions retrieves the information carried in the message and enforces the reservation *b* of the control CoS, and the initial over-reservations for each CoS on each of its interfaces I_e using the equation (3.14). After that, the node records the ID of the previous node visited by the message in its local MRIB to avoid asymmetric routing in the reverse direction. Then, it forwards a copy of the message on each of its interface (except the one on which it was received) after appending the capacity and the ID of the interface to the Route Record Object carried in the message. A node does not append its initial reservations parameters for simplicity, the CDPs computes them since they include the functions of the core nodes and the interfaces capacities are collected. Every node is enabled to discard loop messages by checking the list of interfaces' IDs carried in each message, and the initial reservations are also enforced only once to avoid duplications. This process is repeated on every node visited until a message is received by a remote CDP.

Hence, when a CDP receives a message which was not initiated by itself, it creates a corresponding response message and sends it back in the reverse route to the CDP which initiated the message. A response message carries the information collected by the original message, such as the list of interfaces' IDs and the corresponding list of capacities. This way, a CDP creates and collects key information on all the possible routes from itself to other CDPs inside the network. Afterwards, the CDP assigns a multicast channel per route and signals the latter to enforce the channel, thus transforming each route into bandwidth-aware multicast tree (with the initial over-reservations configured). After a CDP has created all its possible trees, the QoS-aware trees, it selects the best trees which can be exploited for sessions transport during the system running time. For simplicity in this Thesis, a CDP selects its trees based on the number of hops and the available bandwidth on the bottleneck outgoing interface of the tree. To improve system throughput and resource utilization, a CDP is allowed to maintain multiple trees and the remained trees are kept for use upon need for system robustness. However, for simplicity in this illustrative description using Figure 4.3 in step (a), a CDP simply selects a single unbranched tree to connect each of the remote CDPs.

4.1.1.2 Initial Synchronization Control Functions

The initial synchronization functions mainly enable each CDP to build its knowledge of the initial status of the network. For this purpose, after a CDP has selected its useful trees, it invokes the initial Synchronization Functions to advertise other CDPs about its selected trees IDs, and to advertise the non-selected also with the list of outgoing interfaces on each tree. This way, the CDPs exchange their trees and the related key information in *step (b)* on Figure 4.3, thus allowing every CDP to be aware of all trees created (selected and non-selected) in the network. Then, every CDP builds its own NetCIB control database to store network topology and appropriate control data (see *step (c)* on Figure 4.3). A CDP stores its own selected trees and the related control information in the TREES table (see Table 4.1) in its NetCIB. This information includes the trees indexes, the multicast channel of each tree (to mark packets merged on a tree), and the list of interfaces IDs on each tree. The TREES table stores also the total amount of bandwidth that the CDP grants to all active flows in each CoS_i on each of its tree T_x, denoted as $U_{BW}\left(i,T_x\right)$. In addition, it stores a list of correlated CDPs for each tree, which indicates the CDPs whose trees happen to share outgoing interfaces with that particular tree. As we referred earlier, the list of correlated CDPs of a tree is very important, since it allows a selective cooperation among CDPs for dynamically minimizing the overhead: the resource status on the outgoing interfaces of a given tree is exchanged among the correlated CDPs only.

Besides, the CDP records the outgoing interfaces IDs of its selected trees and related key

control information in its TOPOLOGY table (see Table 4.2). This information includes the interface's ID (e.g., I_e), interface capacity, the interface sharing factors (number of selected trees that use the interface), the amount of bandwidth reserved for each CoS_i on each interface I_e, denoted as $R_{BW}(i, I_e)$, the total amount of used bandwidth in each CoS_i. The IDs of the selected trees that use the interface are also recorded together with the ID of the CDP to which the tree belongs, which is the Tree Correlations Pattern on the concerned interface.

Table 4.1. TREES table.

Tree index	Egress CDP_ID	Multicast channel	EF Used	AF Used	BE Used	*Outgoing interfaces*	Correlated CDPs per tree
0	CDP5	(CDP1, CDP5)	0	0	0	I_1; I_4	CDP2; CDP3; CDP4
1	CDP4	(CDP1, CDP4)	0	0	0	I_2; I_9; I_{14}	CDP2; CDP3; CDP5
2	CDP3	(CDP1, CDP3)	0	0	0	I_2; I_{10}; I_{26}	CDP2; CDP5
3	CDP2	(CDP1, CDP2)	0	0	0	I_2; I_{11}	CDP5

Table 4.2. TOPOLOGY table.

Interface ID	Interface capacity	Interface sharing factor	CS	EF		AF		BE		Trees correlations patterns (*CDP_ID: Tree_Index*)
				Rsv	Total used	Rsv	Total used	Rsv	Total used	
I1	1000	1	1	399.6	0	299.7	0	299.7	0	(1: 0)
I2	1000	3	1	399.6	0	299.7	0	299.7	0	(1: 1); (1: 2); (1: 3)
I4	1000	4	1	399.6	0	299.7	0	299.7	0	(1: 0); (2: 1); (3: 2); (4: 3)
I9	1000	1	1	399.6	0	299.7	0	299.7	0	(1: 1)
I10	1000	1	1	399.6	0	299.7	0	299.7	0	(1: 2)
I11	1000	2	1	399.6	0	299.7	0	299.7	0	(1: 3); (5: 2)
I14	1000	4	1	399.6	0	299.7	0	299.7	0	(1: 1); (2: 2); (3: 3); (5: 0)
I26	1000	3	1	399.6	0	299.7	0	299.7	0	(1: 2); (2: 3); (5: 1);

Finally, the VOPRS table (Table 4.3) is used to store the VOPR of each of the CDP's selected tree for each CoS on each outgoing interface composing the tree. A VOPR in a CoS_i for a tree T_x on an interface I_e, $Vopr(i, I_e, T_x)$, is a share of the reserved bandwidth on the CoS_i for the tree T_x on that interface, and is given by:

$$Vopr(i, I_e, T_x) = U_{BW}(i, T_x) + \frac{R_{BW}(i, I_e) - U_{BW}(i, I_e)}{Factor(I_e)} \tag{4.1}$$

where $R_{BW}(i, I_e)$ is the current reservation of the CoS_i on the interface I_e, *Factor(I_e)* is the sharing factor of I_e which is the number of trees that share I_e (see Table 4.2), $U_{BW}(i, T_x)$ is the sum of the amount of bandwidth $\bar{r}_i^f(T_x)$ granted to each active flow *f* mapped in CoS_i onto the path T_x (see Table 4.1), and $U_{BW}(i, I_e)$ is the total amount of bandwidth granted to the active flows (on all

trees) in the CoS_i through I_e (see Table 4.2). Hence, $U_{BW}(i,T_x)$ and $U_{BW}(i,I_e)$ are respectively obtained using the following equations (4.2) and (4.3):

$$U_{BW}(i,T_x) = \sum_{f=1}^{\Phi} \bar{r}_i^{\,f}(T_x) \tag{4.2}$$

$$U_{BW}(i,I_e) = \sum_{x=1}^{m}\sum_{f=1}^{\Phi} \bar{r}_i^{\,f}(T_x) \tag{4.3}$$

where m is the total number of trees that use the interface I_e and Φ is the total number of flows mapped in CoS_i onto each path T_x.

Therefore, a CDP can admit multiple flows in a CoS_i on a tree T_x without requiring synchronization as long as the related VOPR is available. As a result, the VOPR concept provides means to hide cross-traffic loads dynamics from remote CDPs, while sharing resources and allows for improving performance. To prevent CoS starvation and waste of VOPRs, the SC functions are responsible for dynamically readjusting the VOPRs in a way that properly redistributes the allocated but unused VOPRs in correlated trees upon need, as we further detail in subsequent subsection.

Table 4.3. VOPRS table.

Tree index	*Interface_ID*: ($VOPR_{EF}$; $VOPR_{AF}$; $VOPR_{BE}$)
0	I_1: (399.6; 299.7; 299.7); I_4: (99.9; 74.925; 74.925)
1	I_2: (133.2; 99.2; 99.2); I_9: (399.6; 299.7; 299.7); I_{14}: (99.9; 74.925; 74.925)
2	I_2: (133.2; 99.2; 99.2); I_{10}: (399.6; 299.7; 299.7); I_{26}: (133.2; 99.9; 99.9)
3	I_2: (133.2; 99.2; 99.2); I_{11}: (199.8; 149.85; 149.85)

4.1.2 **ACOR Operations at Network Running Time**

The ACOR operations at network running time consist on the interactions between the ACOR components to assure proper session admission, and the transport in a network with QoS guarantees, and at low control cost in terms of signalling and related overhead reduction without incurring in QoS violations or waste of bandwidth. The resource and admission process of ACOR can be classified into two phases based on the VOPR availability: 1) *Resource and Admission without Signalling Phase* (RAoutS), consists of processing sessions without synchronization or QoS reservation signalling into the network as long as requested VOPRs are available in the network; 2) *Resource and Admission upon Signalling Phase* (RAuS), which encompasses four steps: a) Tringgering Synchronization, which triggers cooperation between CDPs for synchronization; b) Successful Admission without Reservation Signalling; c) Successful Admission

upon Reservation Signalling; and d) Unsuccessful Admission, where thre request is denied due to insufficient resource availability. To facilitate the understanding, the subsections 4.1.2.1, 4.1.2.2, and 4.1.2.3 describe these operations with respect to the AC, SC, and RC and BF components respectively, based on Figure 4.4.

4.1.2.1 Admission Control Functions

When an ingress CDP_A receives an authorized multicast session request r_i to a CoS_i and destined to a given egress CDP_B in the control domain (see Figure 4.4), the CDP_A processes the flow admission control (*Phase I)* as in the following. First, it collects the candidate trees of the incoming request, being its own trees that connect to the desired egress CDP_B. Then, among the candidate trees, it selects the one (e.g., T_x) which has the highest available VOPR denoted as $A_{Vopr}(i,T_x)$, that is, ($r_i \le A_{Vopr}(i,T_x)$). It is important to note that an available VOPR $A_{Vopr}(i,T_x)$ of a CoS_i ($1 \le i \le k$) on a tree T_x is the amount of VOPR of the tree which has not been allocated to any flow on its bottleneck outgoing interface, and is obtained using the following function:

$$A_{Vopr}(i,T_x) = \min\left\{Vopr(i,I_e,T_x) - U_{BW}(i,T_x)\right\} \tag{4.4}$$

where I_e is an outgoing interface on the tree T_x, $Vopr(i,I_e,T_x)$ is obtained from equation (4.1), and $U_{BW}(i,T_x)$ is obtained using the equation (4.2).

If the admission is successful ($r_i \le A_{Vopr}(i,T_x)$), CDP_A maps the request to the CoS_i on that tree T_x without synchronization or reservation readjustment signalling event. The same way, a CDP does not trigger synchronization or reservation readjustment signalling event when it releases/terminates an active flow from a CoS_i on a tree T_x. Hence, after admitting a new flow or releasing an active flow in a CoS_i on a tree T_x, a CDP updates the used bandwidth statistics $U_{BW}(i,T_x)$ of the concerned CoS_i and tree T_x in its NetCIB, according to the used bandwidth of the flow, that is, ($U_{BW}(i,T_x) \leftarrow U_{BW}(i,T_x) \pm r_i$) – (see Table 4.1.)

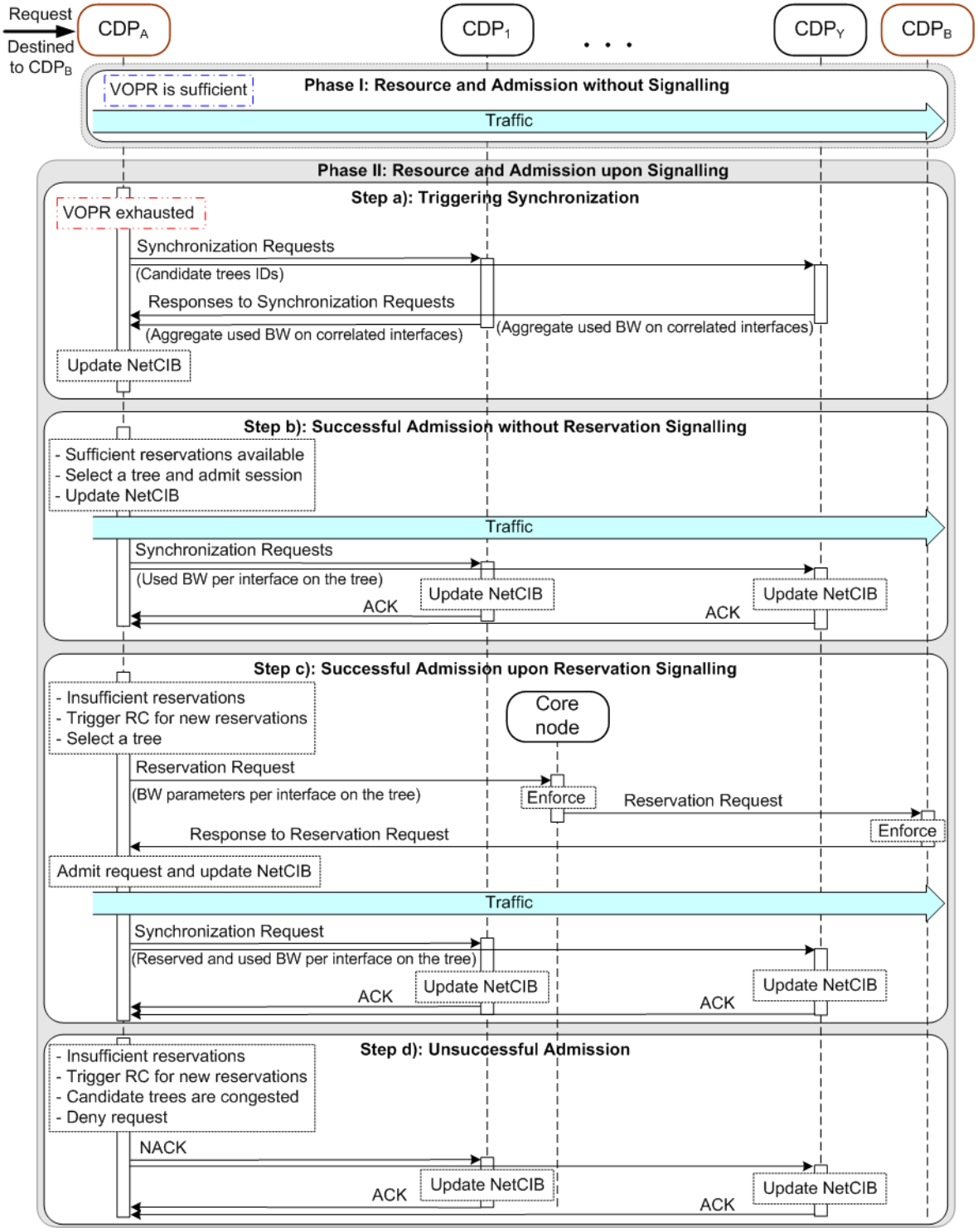


Figure 4.4. Illustration of ACOR messages sequence chart.

In addition to updating the local database, traffic conditioning (e.g., flows shaping and policing) is enforced at each ingress CDP, thus forcing active flows to comply with the contracted behavior during QoS negotiation control. Also, the packets of each admitted session are pinned to the multicast tree mapped to the session, so that they enjoy the QoS destined to them on the tree, while allowing the CDPs to maintain consistent data information at the edge without excessive

signalling into the network. Thus, each CDP maintains information of the available VOPRs in each CoS on every outgoing interface in real-time based on equation (4.4) ($Vopr(i,I_e,T_x)-U_{BW}(i,T_x)$) to avoid QoS violation and waste of resources.

However, in case the VOPR is insufficient on the bottleneck outgoing interfaces of all its candidate trees ($r_i > A_{Vopr}(i,T_x)$) using the equation (4.4), the operations Phase II (see Figure 4.4) is triggered as it is detailed in the following.

4.1.2.2 Synchronization Control Functions

The synchronization control functions are triggered in *step (a)* in Figure 4.4 uon the requested VOPR exhaustion so that the network overall resource status of each CoS can be properly updated to prevent VOPR starvation or waste of resource. To this end, the CDP_A collects the correlated CDPs (e.g., CDP_1 and CDP_Y) of the candidate trees from its TREES table (see Table 4.1) and advertises the latter with the IDs of the candidate trees in a control message. This way, every CDP exploits the list of correlated CDPs dynamically to assure selective cooperation. Then, each remote CDP retrieves the common interfaces between its own and the candidate trees conveyed in the message based on the tree correlations patterns in its TOPOLOGY table (see Table 4.2). Let *h* be the number of interfaces stored in the TOPOLOGY table, *m* the number of candidate trees, and $L_{Pattern}(I_e)$ the list of trees' IDs which compose the correlations pattern on a given interface I_e. Thus, given two integers *e* and *x*, the process can be the following:

Algorithm 4.1: Common interfaces between correlated trees and the own trees.

```
/*For each interface I_e in the TOPOLOGY.*/
for e=1:h do
    /*For each candidate tree T_x. */
    for x=1:m do
        /* If candidate tree T_x in pattern. */
        if (T_x ∈ L_Pattern(I_e)) then
            (a) Record I_e as a common interface;
            (b) Record associated own trees from L_Pattern(I_e);
        end
    end
end
```

This way, each CDP obtains a list of the common interfaces, together with the IDs of own correlated trees per interface. After that, it sets the own correlated trees on the common interfaces to *Standby Mode*. The *Standby Mode* is a means to carefully prevent admitting new flows on a tree when the resource reservation parameters are being readjusted on the interfaces that belong to the tree, which is important to avoid QoS violation. Hence, it affects only the requests that see VOPR exhaustion on their desired trees; otherwise, the requests are processed without signalling as

explained earlier in subsection 4.1.2.1. After that, the CDP computes the aggregate used bandwidth $U_{AggrBW}(i, I_e)$ of its own trees in each CoS_i on each common own interface I_e, the aggregate used bandwidth on I_e, as:

$$U_{AggrBW}(i, I_e) = \sum_{x=1}^{m'} U_{BW}(i, T_x) \tag{4.5}$$

where, *m'* is the number of the own trees that share the common interface I_e.

Then, the CDP encapsulates its aggregate used bandwidth in each CoS in a control message, and sends it to the CDP_A which initiated the synchronization. Thus, the CDP_A collects the aggregate used bandwidth statistics of every correlated CDP on the outgoing interfaces of the candidate trees, and therefore computes the total amount of used bandwidth in each CoS on each of the interfaces. Then, it updates the total used bandwidth statistics of the interfaces in its TOPOLOGY table accordingly. Afterwards, it triggers the AC functions in *step (b)* to process the incoming requests. AC first checks the available reservation $A_{RservBW}(i, T_x)$ (the over-reserved but unused) in requested CoS_i on the bottleneck interfaces of the candidate trees as:

$$A_{RservBW}(i, T_x) = \min\{R_{BW}(i, I_e) - U_{BW}(i, I_e)\} \tag{4.6}$$

where I_e is an outgoing interface on a candidate tree T_x.

Thus, in case the admission succeeds on a candidate tree ($r_i \leq A_{RservBW}(i, T_x)$), the CDP admits the flow on the candidate tree which has the highest available over-reservation (load balancing) without QoS reservation readjustment on any candidate tree.

After the flows have been admitted, CDP_A updates the used bandwidth statistics in requested CoSs on the used tree(s) in its TREES table, according to the amount of the bandwidth granted to the new flows r_i. Besides, it updates in its TOPOLOGY table the total amount of used bandwidth statistics of the requested CoS_i on each outgoing interface, which lies on the candidate tree(s) used for the admission. Also, it updates the VOPRs of each CoS_i on each outgoing interface that belongs to the candidate trees used to admit the flows, in its VOPRS table (see Table 4.3) using the equation (4.1). Finally, CDP_A encapsulates the updated total used bandwidth statistics of each CoS_i on the outgoing interfaces of the candidate trees used to admit flows in control messages, and sends them to each of the other correlated CDPs. Hence, each remote CDP also updates its local database accordingly and resets its correlated trees to *Normal Mode*. Finally, it acknowledges CDP_A of the receipt of the message. This way, CDPs are enabled to selectively cooperate in a way to properly update their local databases upon need to avoid QoS violations and VOPR resource starvation.

However, in case the admission control cannot succeed using the function in equation (4.6), the CDP triggers the RC functions in *step (c)* to allow for readjustment of the residual bandwidths among CoSs to avoid CoS starvation, waste of resources and unnecessary increase of session blocking probability. The RC functions are detailed in subsection 4.1.2.3.

It is also assured a proper process of synchronization when several CDPs happen to trigger synchronization events simultaneously over the same correlated trees, through a unique priority tag. In case several events show incoming requests with the same priority, the CDP with the smallest ID is automatically elected.

4.1.2.3 Resource Control Functions

When the RC funtions are invoked in *step (c)* in Figure 4.4, the candidate trees to process are set to *Standby Mode*. Then, RC component implementing ECOR algorithm computes the total unused bandwidth $\Delta_T\left(I_b,T_x\right)$ on the bottleneck outgoing interface I_b of each candidate tree T_x as:

$$\Delta_T(I_b,T_x)=\min\{\sum_{i=1}^{k}(R_{BW}(i,I_e)-U_{BW}(i,I_e))\} \tag{4.7}$$

where $R_{BW}\left(i,I_e\right)$ and $U_{BW}\left(i,I_e\right)$ are respectively the reservation and used bandwidth of each CoS_i $(1\le i\le k)$ on each outgoing interface I_e which lies on the tree T_x. While expression (4.6) provides the smallest amount of unused bandwidth in a CoS_i along a tree T_x, the expression (4.7) refers to the smallest amount of the total unused bandwidth (of all CoSs) on the outgoing interfaces that belong to a tree T_x.

In case the function in equation (4.7) succeeds ($r_i \le \Delta_T\left(I_b,T_x\right)$) on a candidate tree T_x, the CDP selects the tree T_x which has the highest unused resources on its bottleneck outgoing interface, and ECOR algorithm is triggered to define new reservation parameters for each outgoing interface I_e on the selected tree T_x. Hence, besides ECOR, any over-reservation algorithm (e.g., COR or MARA detailed in Chapter 3) could be thus deployed to compute new reservation parameters in ACOR.

This way, ACOR obtains new reservation parameters for every CoS on each outgoing interface of a tree based on the resource conditions on the interface itself. Once the new reservation parameters are successfully defined, they are encapsulated in appropriate QoS objects inside a reservation control message and conveyed to the nodes on the tree. Hence, as the message is travelling along a tree, each node, hosting the Basic Functions, intercepts the message and retrieves the parameters destined to its local outgoing interface on the tree. Then, the node enforces the new configurations on the interface accordingly, so that each CoS receives the amount of resources allocated to it. Then, the egress CDP_B creates a response message and acknowledges the ingress

CDP_A which initiated the reservations. After that, the AC functions are triggered and CDP_A admits the request and updates the used bandwidths in its TREES and TOPOLOGY tables. After that, it updates the VOPRs of each CoS on each of the interfaces that compose the candidate tree(s) readjusted, and resets its candidate paths and the correlated ones to *Normal Mode*. Finally, the SC module is triggered to encapsulate the updated used bandwidth, and the new reservations of each CoS on the outgoing interfaces that compose the candidate trees.

If the incoming flow request cannot be admitted on any candidate tree ($r_i > \Delta_T\left(I_b, T_x\right)$), the network is said to be congested, and CDP_A denies the request. In this case, CDP_A resets its candidate trees and the correlated ones to *Normal Mode* and acknowledges the correlated CDPs. Hence, upon receiving the acknowledgement, each CDP resets its correlated trees to *Normal Mode* and acknowledges the CDP_A, and the control process resumes to normal. The signalling protocol developed to support the mechanisms designed in this Thesis is described in the following section 4.2.

4.2 *ACOR Control Signalling Protocol*

The ACOR-P is an NSIS compliant signalling protocol, which provides support for the overall control mechanisms designed in this Thesis. In particular, ACOR-P extends the generic functionalities of the QoS-NSLP signalling protocol with specific control parameters, QSPECs, Synchronization Context Information Specification (CXT_SPEC) objects, and appropriate message processing rules to achieve proper operations of the ACA described in Chapter 3 and the ACOR in this chapter, for example. The main objective of this section is to provide an overview of the ACOR-P common objects, the QoS and synchronization control object specifications, including the message structures and transport mechanism, knowing that further detailed on the message fields, format, and types and values are provided in appendix.

4.2.1 **ACOR-P Common Objects**

In terms of control signalling message fields and objects, ACOR-P reuses the *Common Header*, detailed in QoS NSLP [123], in order to define message type and the general control specification flags. It also uses the *Request Identification Information* (RII) object to properly associate a response message to its corresponding original (e.g., reserve) message, since the message type uniquely identifies each particular message. Further, the Record Route Object (RRO) is exploited to collect sequential list of the visited nodes' IDs (e.g., IP or MAC addresses of the nodes'outgoing interfaces) on the communication paths (e.g., in ACA and ACOR initialialization phase). To properly report feedback on control operations (e.g., error, failures, etc.), the Information

Specification object (*INFO_SPEC*) has been extended while multicast operations are supported by means of the *Multicast Specification* (MSPEC) object introduced in [125]. The NSLP QSPECs objects detailed in [123] have also been adapted according to the requirements of the QoS model designed in this Thesis (e.g., ACA and ACOR QoS Model). Further, ACOR-P introduces CXT_SPEC objects to provide support for dynamic synchronization between CDPs, scalable QoS and survivability control. The QSPEC, CXT_SPEC, Message structures and message transport are introduced in the following.

4.2.1.1 QSPEC Specification

The QSPEC object is used to carry the QoS information required on each relevant outgoing interface on trees to assure the QoS provisioning approach proposed in this Thesis. As one can see in Figure 4.5, an ACOR-P QSPEC is tagged with the ID of the interface to which the QoS parameters carried are destined. This is very important in our designs, since the QoS control on an interface depends on the resource conditions on the interface, and one message may convey different information to different nodes along a tree. Hence, as a message is travelling along a tree, each node on the tree is enabled to intercept it, and to retrieve the object destined to its local interfaces so as to take proper processing actions accordingly. Basically, an ACOR-P QSPEC object contains a common header, the ID of the corresponding outgoing interface, along with the appropriate QoS objects (e.g., QoS desired, QoS available, etc.). As it is specified in QoS-NSLP protocol, each QoS object is composed of an object header and the object parameters, while each parameter consists on a parameter header and the parameter itself, respectively as illustrated in Figure 4.5. Further details on the fields and format are provided in the appendix B.

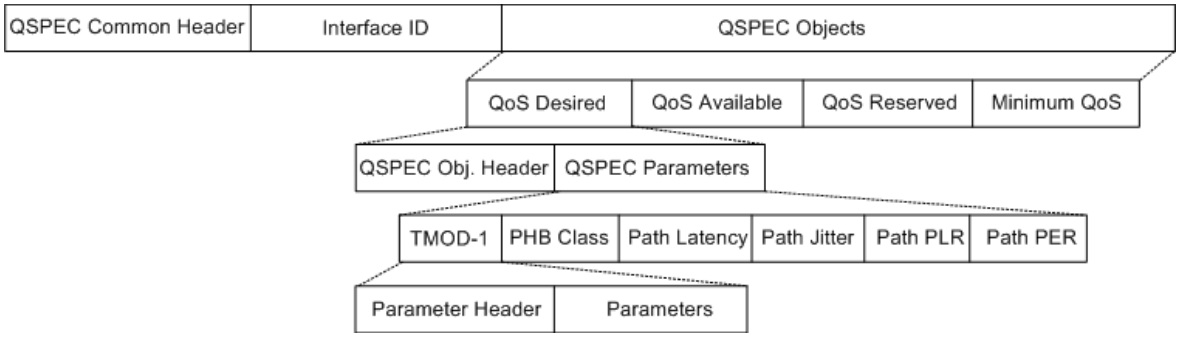


Figure 4.5. ACOR QSPEC structure.

4.2.1.2 Synchronization Context Information Specification (CXT_SPEC)

The synchronization CXT_SPEC object, illustrated in Figure 4.6, provides support for the synchronization and survivability control mechanisms designed in this Thesis. As being an NSIS compliant protocol, the CXT_SPEC object uses the same Type Length Value (TLV) format as the QSPEC object, and each CXT_SPEC object (e.g., VOPR, Survivability) has a common header

along with the CXTSPEC objects. For example, the VOPR object is used to convey the information exchanged between CDPs for the purpose of synchronization, while the Survivability object is used in the context of network survivability control as in Chapter 6. This way, each CXTSPEC object is composed by its own Object header and the context information parameters to be exchanged between CDPs. Further, each context parameter is made of the parameter itself together with the parameter header which structure is further described in appendix B.

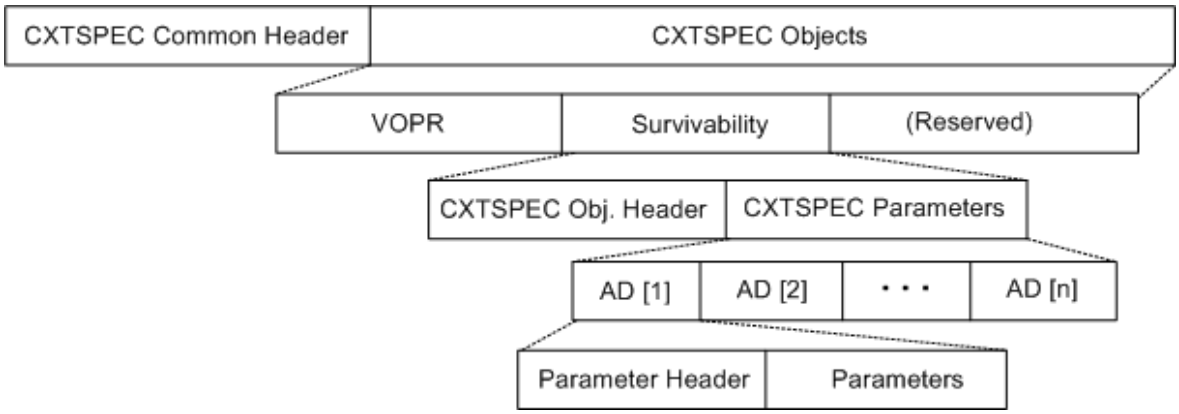


Figure 4.6. ACOR CXTSPEC structure.

4.2.2 **ACOR-P Signalling Message Genercic Structure**

A generic structure of the ACOR-P control message is illustrated in Figure 4.7. In particular, it reuses some of the generic functionalities provided by QoS-NSLP [123] such as Common Header (to define message type and general control specification flags), Message ID (to uniquely identify messages), the Request Identification Information (to correctly associate a response message to the original message), and information specification (to report feedback about control operations). In addition, a control message (e.g., reserve message, synchronization message, response message, etc.), identified by the message type, may carry specific control information according to the operations in course. This information includes Initiator QoS Specification (to describe original traffic and QoS characteristics from media source), Local QoS Specification (converted Initiator QSPEC into local QSPEC in heterogeneous QoS Models environment). A message may also carry Multicast Channel Specification to be enforced on routes for media transport, Record Route Object as topological and route information, while the Synchronization Context Information Specification is used to assure proper decentralization of the control. For the sake of simplicity, we only define two types of messages: *REQUEST* message which is used to initiate a communication, and the *RESPONSE* message which is used as the corresponding feedback. Hence, different message types (e.g., QoS reservation, synchronization, survivability, multicast tree control, etc.) are identified by means of appropriate control flags.

Common Header	Message ID	RII	INFOSPEC
Initiator QSPEC *	Local QSPEC *	MSPEC (Multicast specification) *	
Record Route Object (RRO) *	CXTSPEC (Context Information Specification) *		

* Optional

Figure 4.7. ACOR-P messages generic structure.

4.2.3 ACOR-P Signalling Message Transport

As we illustrate in Figure 4.8 using the ACOR architecture described in this chapter, ACOR-P messages are transported in UDP datagram with UDP port recognition in routers (as routers are permanently listening on UDP port), or Router Allert Option (RAO) [122] is activated which allows routers for capturing the control messages. It is assumed that reliability of the control messages transport is assured, since all control packets are mapped to bandwidth-aware and dedicated control CoS, and a feedback is expected for all control messages. In this sense, Figure 4.8 illustrates the transport of a QoS reservation message from an ingress node to an egress node inside a single network control domain, where the CDPs at the edge are connected to external users represented by a media source and a media destination. Hence, the ingress node generates a QoS reservation message, encapsulates it in a UDP datagram in step 1, and sends it down the network in step 2. When the message is captured by a core router in step 3, the latter interprets and processes the message according to the control instructions conveyed in the message. Afterwards, the message is encapsulated in step 4 and forwarded in step 5. This procedure is repeated at every visited core node until the message is received by the egress node in step 6. Hence, after processing the message, the egress node encapsulates the corresponding response message in step 7, and sends it back in step 8 to the ingress node which initiated the process. Then, the message is forwarded as in step 9 until it is received by the ingress node in step 10.

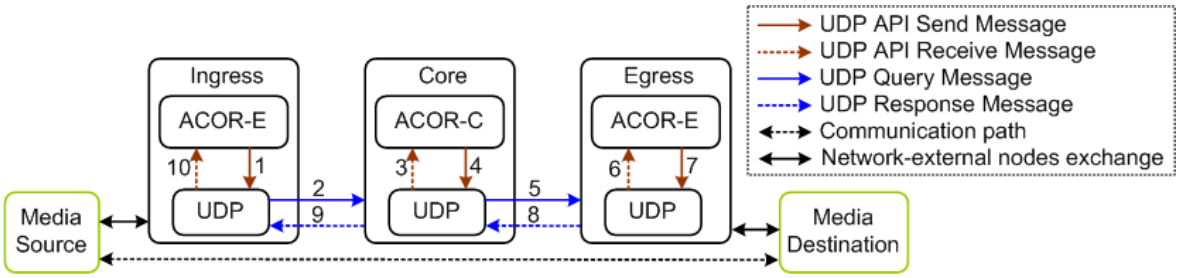


Figure 4.8. ACOR-P messages transport.

The subsequent section provides analytical model for generic purpose assessment of ACOR, especially the influence of various control parameters (e.g., sessions, dynamics, link capacity, etc) on the performance of resource overprovisioning approach in general.

4.3 *ACOR Analytical Model*

Figure 4.9 is used to illustrate bottleneck scenarios and to provide analytical model of the proposed approach. For this purpose, it is assumed that *m* trees (T_1, … , T_x, … , T_m) originated respectively from the CDP_1, … , CDP_X, … , CDP_Z happen to share a bottleneck outgoing interface I_b in a network. Besides, it is considered that each interface in the network has a capacity *C* and implements *k* CoSs (e.g., EF, AF and BE) and one control CoS.

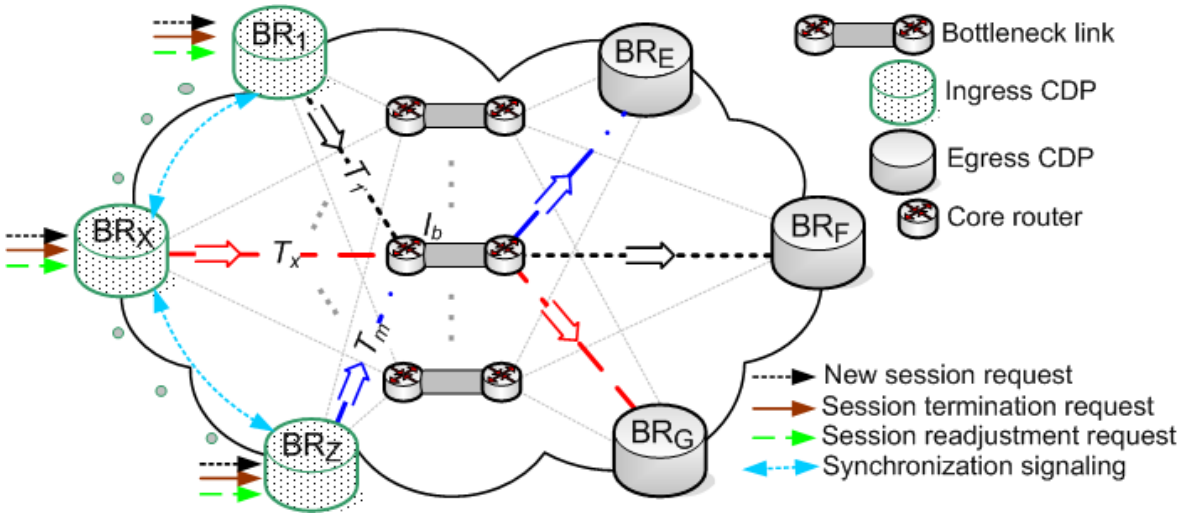


Figure 4.9. Topology for analytical study.

In order to facilitate the understanding of this study, Figure 4.10 illustrates the description using a bottleneck outgoing interface of a node **A** towards a node **B**.

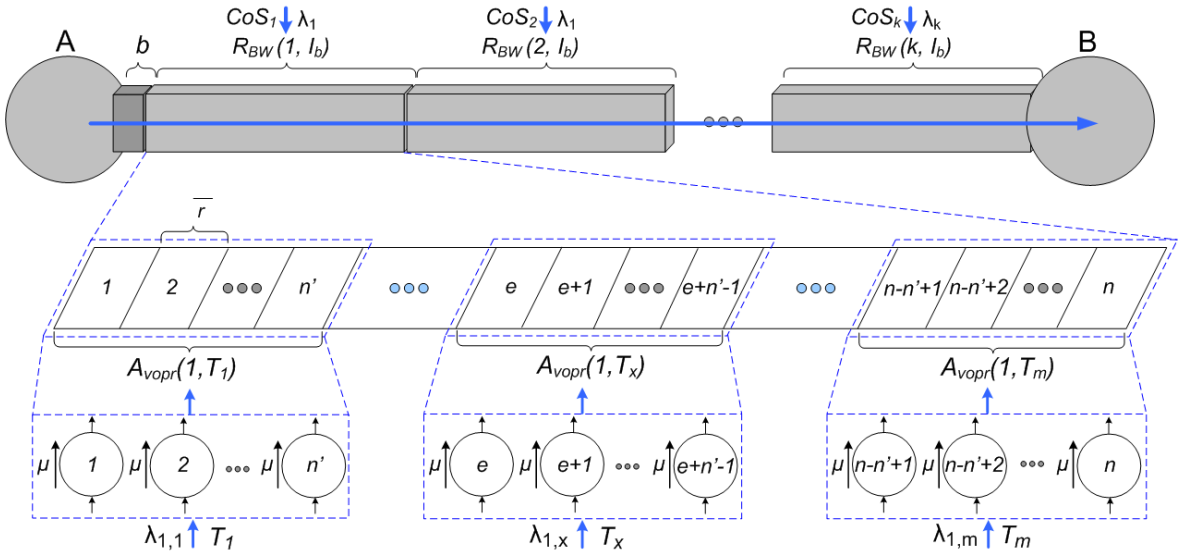


Figure 4.10. Proposed control model.

Hence, the *m* trees (T_1, …, T_x ,…, T_m) share the interface (**A→ B**), and an amount of bandwidth $R_{BW}(i, I_b)$ is over-reserved for each CoS_i with $(1 \le i \le k)$ on the interface. Flow connection

requests to a CoS_i on the trees that use the interface (**A→ B**) are Poisson processes with rate λ_i, and the requests to a CoS_i on a given tree T_x that use the interface (**A→ B**) are Poisson processes with rate $\lambda_{i,x}$ such that:

$$\lambda_i = \sum_{x=1}^{m} \lambda_{i,x} \tag{4.8}$$

4.3.1 ACOR Synchronization Control Model

By considering that m trees share a bottleneck interface I_b (**A→B**) as in Figure 4.10, the surplus or available VOPR $A_{Vopr}(i,T_x)$ of a CoS_i for a given tree T_x with $(1 \le x \le m)$ within a given surplus of over-reserved bandwidth $R_{BW}(i,I_b)$ on the interface is obtained by:

$$A_{Vopr}(i,T_x) = \frac{R_{BW}(i,I_b)}{m} \tag{4.9}$$

Thus, the total number n of sessions that an available VOPR $A_{Vopr}(i,T_x)$ of a CoS_i can accommodate simultaneously without requiring synchronization event for a given tree T_x sharing the interface can be obtained by:

$$n = \left\lfloor \frac{R_{BW}(i,I_b)}{r_i * m} \right\rfloor \tag{4.10}$$

Therefore, an available VOPR $A_{Vopr}(i,T_x)$ of a CoS_i for a given tree T_x is characterised by:

- n possible sessions slots of a mean bandwidth r_i each are over-allocated for each tree T_x in CoS_i as $A_{Vopr}(i,T_x)$ within $R_{BW}(i,I_b)$;
- Session requests to a CoS_i on a tree T_x are Poisson processes with rate $\lambda_{i,x}$;
- Session's lifetime is exponentially distributed with mean τ;
- Session requests arrival process is independent of the session lifetime (service time);
- An available VOPR $A_{Vopr}(i,T_x)$ can only accommodate n sessions simultaneously without requiring synchronization.

The VOPR control approach is then modelled as an M/M/n/n queuing system as depicted in Figure 4.10. Thus, the probability $P_{i,x}$ that the available VOPR $A_{Vopr}(i,T_x)$ of a tree T_x in a CoS_i exhausts to trigger synchronization is the probability that an incoming request to the CoS_i on the

tree T_x finds all the n "VOPRed sessions" slots occupied. Therefore, it can be obtained using Erlang B formula as:

$$P_{i,x} = \frac{\left(\frac{\lambda_{i,x}}{\mu}\right)^{n'} * \frac{1}{n'!}}{\sum_{\alpha=0}^{n'} \left(\frac{\lambda_{i,x}}{\mu}\right)^{\alpha} * \frac{1}{\alpha!}} \tag{4.11}$$

From the equations (4.9), (4.10), (4.11), one can see that the frequency of synchronization events depends on several parameters such as interfaces' capacity C, resource utilization level, mean bandwidth r_i allocated to each session, session requests arrival rate λ_i to a CoS_i, session requests arrival rate $\lambda_{i,x}$ to CoS_i on the tree T_x sharing the bottleneck interface, session mean lifetime τ, the number of CoSs implemented on interfaces, and the number of trees that share the bottleneck interface. It also depends, as we observed in Chapter 3, on the over-reservation algorithm in use (COR, ECOR or MARA).

4.4 *Performance Evaluation*

The benefits of ACOR control mechanism (ACOR architecture embedding ECOR – ECOR/ACOR) were evaluated through analytical study and simulation analysis using the ns-2. To compare results, we embedded the resource over-reservation computation algorithms of COR and the competing state-of-the-art MARA in the ACOR RC module described earlier in section 4.1, thus allowing each of the algorithms to benefit from the ACOR Admission and Synchronization Control functions to prevent QoS violation. Hence, while ACOR refers to the ACOR architecture embedding ECOR algorithm, the terms COR and MARA will be used to refer respectively to ACOR architecture embedding COR and MARA algorithms, in order to ease the understanding of our description in the rest of this section. Regarding the analytical results using the over-reservation algorithm of COR, ECOR and MARA, our analysis bases on the same assumption detailed in Chapter 3 - section 3.5.1 and therefore will not be repeated here.

4.4.1 Analytical Results

Based on the configurations in Table 4.4, Figure 4.11 plots the probability of both the synchronization and the reservation signalling events occurrence as a function of network bottleneck interface's resource utilization level on communication trees. The reservation results are obtained using the over-reservation control model and assumptions described in Chapter 3.

Table 4.4. Configuration parameters for resource utilization level scenario.

$m = 3$	Bottleneck interface sharing factor
$k = 3$	Number of service CoSs implemented on the interface
$\bar{r} = 1$	Mean bandwidth requested by each session (Mbps)
$\mu = 1/4$	Mean service rate per session (requests/time unit)
$\lambda_i = 20$	Session requests arrival rate to a CoS_i (requests/time unit)
$\lambda_{i,x} = 9$	Arrival rate of session requests to a CoS_i on a tree T_x
$C = 1000$	Interface capacity (Mbps)

As one can see in Figure 4.11, the increase in the unused session slots on the bottleneck interface of a tree decreases the probability of signalling occurrence. The probability of synchronization events is higher than that of reservation events, as expected according to equations (3.19) and (4.10) respectively. Hence, by allowing for over-reserving as much resources as a CoS requires and efficiently reusing the residual resources among existing CoSs dynamically to prevent waste of resources, ACOR outperforms both COR and MARA in terms of probability of signalling events occurrence. Both COR and MARA over-reserve a relatively small portion of unused resources each time in order to reduce performance's negative impacts, since they were not designed with ACOR's knowledge of network topology and related links resource statistics on real-time basis. Besides, MARA is subject to high probability of signalling when Q is smaller than 400, i.e. network is close to congestion, and shows different behavior when Q is higher than 400, in the lower network utilization phase; the same reasons to which we referred in Chapter 3 are applied here. One can also notice that the ACOR reservations are "Not a Number - NaN" when the number of unused slots is 571 or 666 (too high), which implies that resource control would not be necessary if link bandwidth were unlimited.

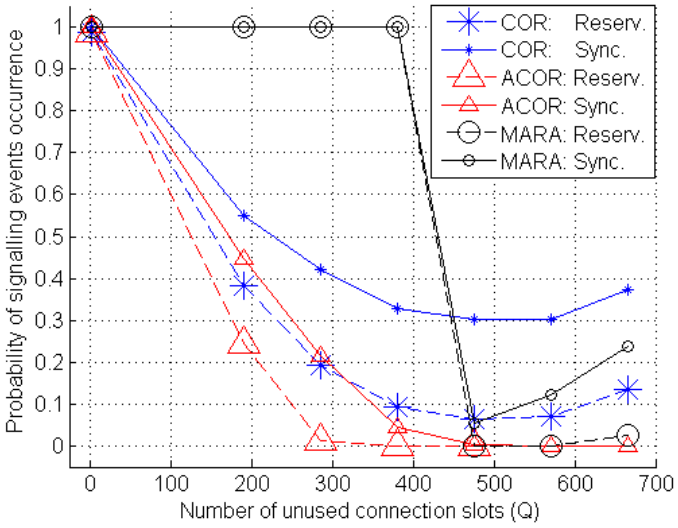


Figure 4.11. Effect of resource utilization level on overall signalling frequency.

Figure 4.12 analyses the effect of session lifetime (e.g., short-lived, long-lived, etc…) and the suitability of resource over-reservation in dynamic network scenarios. In order to clearly observe the effect of session lifetime on performance in terms of signalling occurrence rate, we increase session arrival rates, the number of CoSs and set the number of free slots to 666 as summarized in Table 4.5.

Table 4.5. Configuration parameters for sessions lifetime scenario.

$m = 3$	Bottleneck interface sharing factor.
$k = 8$	Number of CoSs implemented on the interface.
$\bar{r} = 1$	Mean bandwidth requested by each session (Mbps).
$\lambda_i = 70$	Session requests arrival rate to a CoS_i (requests/time unit).
$\lambda_{i,x} = 40$	Arrival rate of session requests to a CoS_i on a tree T_x.
C = 1000	Interface capacity (Mbps).
Q = 666	Interface utilization level.

Hence, we observe in Figure 4.12 that, in a scenario where most of sessions are short-lived (the higher the service rate, the shorter the lifetime), the probability of signalling occurrence is lower than when sessions' lifetime increases. This shows that short-lived sessions leave the reservations more quickly, and these reservations can be reused for accepting other incoming requests without signalling the network.

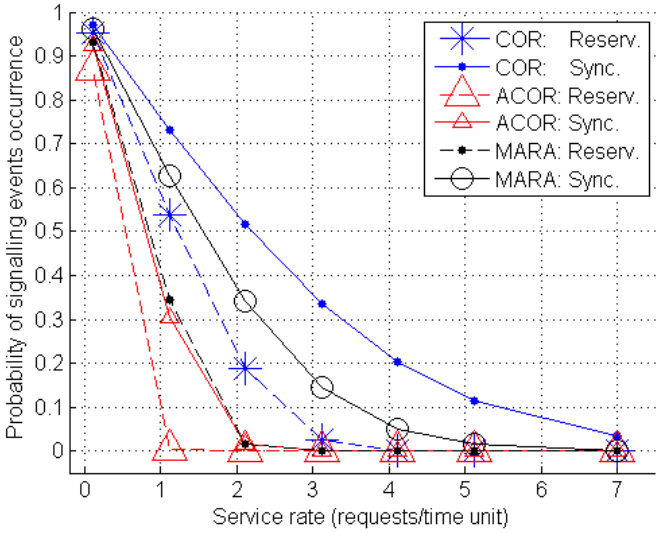


Figure 4.12. Effect of sessions lifetime on overall signalling frequency.

Further, Figure 4.13 analyses the impact of interface sharing factor on the performance. In particular, the probability of synchronization events occurrence increases with the number of trees that share bottleneck interfaces. The higher the number of trees on a bottleneck interface, the more synchronization may be required depending on traffic dynamics on the trees. Besides, it is

important to notice that the interface sharing factor does not affect the reservation signalling events, since reservations are aggregate, and therefore, the readjustment depends on resources exhaustion on the classes. These results could be inferred from equations (3.19) and (4.10). It turns out that appropriate tree filtering techniques allow for further reduction of synchronization events, as described earlier in section 4.1.

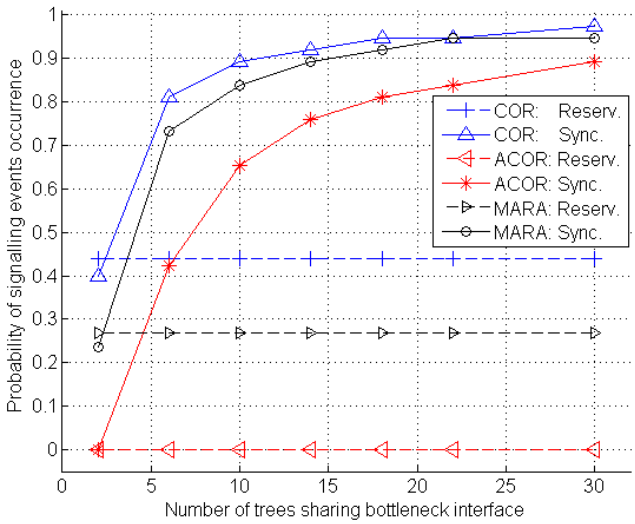


Figure 4.13. Effect of interface sharing factor on signalling frequency.

We analyse also the number of session requests that may be blocked unnecessarily while there are enough unused session slots on the bottleneck interface of a requested tree, and observe the same results as in Figure 3.10 in Chapter 3. In particular, we observe, as in Figure 3.10 that, neither COR nor ACOR blocked any request when there is a free slot, confirming the study in subsection 4.1: COR and ACOR are able to use the sum of residual resources from all existing CoSs to allow for admission of incoming requests until the total unused resources are not enough. However, MARA denied several requests unnecessarily as it is not able to collect all residual resource from existing CoSs, which is crucial when links are close to congestion. These results are confirmed with the simulation results in Figure 4.18.

In summary, we show that it is effectively possible to achieve dynamic aggregate synchronization of multiple control decision points, by means of the VOPR, thus allowing for distributed control in multicast networks, keeping low overall signalling overhead. In general, the more resource is over-reserved for a CoS, the less likely the CoS triggers signalling. However, this must be carefully controlled to prevent waste of bandwidth, which strongly requires a good view on real-time basis of underlying network topology and the related resource status in each CoS.

4.4.2 Simulation Results

In order to obtain results in dynamic and larger scenarios, the over-reservation mechanisms and corresponding architectures were developed in the ns-2 [233]. The simulations were carried out using 4 randomly generated topologies (number of ingress routers ranging from 3 to 6; core routers: 5 to 15, egress routers: 3 to 6) with different degrees of correlations on the links. One of the simulated network topology is presented in Figure 4.14. The obtained results are therefore the mean results of all seeds and topologies.

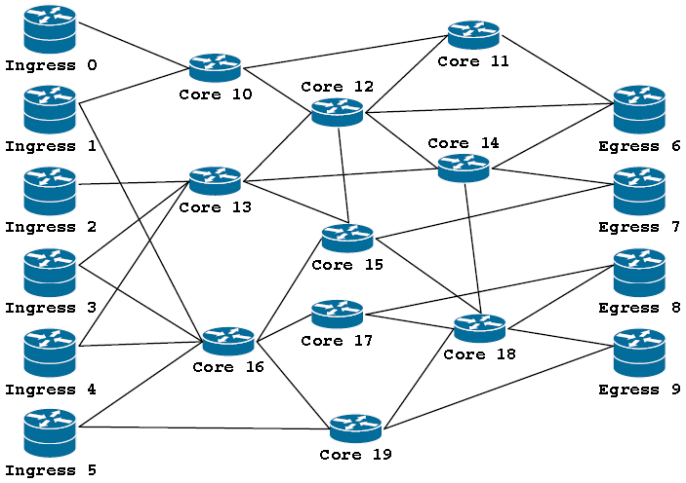


Figure 4.14. Example of simulation network topology.

For simplicity, configurations of CoSs and network interface capacity, and the generation of traffic types and session requests are implemented as in subsection 3.5.3 in Chapter 3

The *network overall resource utilization (%)* in each simulation results' figure is obtained as a mean of the resource utilization level on the bottleneck interfaces of all trees inside the network. The studied metrics include the QoS reservation signalling, the synchronization signalling, the signalling events and load reduction of ACOR in relation to both the COR and MARA, and the unnecessary blocking. To show more accurate results, each simulation is run 10 times with different seeds of random mapping of requests to CoSs, CDPs and egress routers, for each topology. Then, the mean values are plotted for all topologies with a confidence interval of 95%.

Figure 4.15 plots the number of the reservation (Reserv.) and synchronization (Sync.) signalling events and Figure 4.16 plots the corresponding amount of signalling messages load. The messages load is obtained based on the ACOR-P described in section 4.2, which was developed for the overall control mechanism described earlier in section 4.1. As one can see, the reservation signalling events number is lower than that of the synchronization events. More importantly, ACOR clearly outperforms COR and MARA by keeping significantly lower signalling events number and messages load, which confirms our analytical results in sub-section 4.4.1.

Further, we observe that certain data points are null (y = 0), and therefore, are not visible on the graphs due to the log scale plotting. In this sense, one can observe that, after the networks start to operate, synchronization events occur earlier than the reservation events. In particular, around 22% of network overall resource utilization level in Figure 4.15, COR and MARA were already triggering synchronization events, but without any reservation events by then. This demonstrates our two-layer control approach. On one hand, synchronization is triggered upon VOPR exhaustion to allow for proper resource sharing among multiple distributed control decision points. On the other, reservation events occur only when an over-reservation exhausts.

Moreover, the simulation results show similar performance for COR and MARA in terms of number of signalling events and messages load as steadier results. Recall therefore that COR and MARA use the same methods (see equation (3.5) in Chapter 3) to compute surplus of reservation, which is determinant in these performances. Besides, Figure 4.15 and Figure 4.16 do not show a linear relation between the events and the load. Hence, it is important to mention that, the signalling messages triggered by two different signalling events (e.g., reservations) may not carry the same information (e.g., different number of hops on trees).

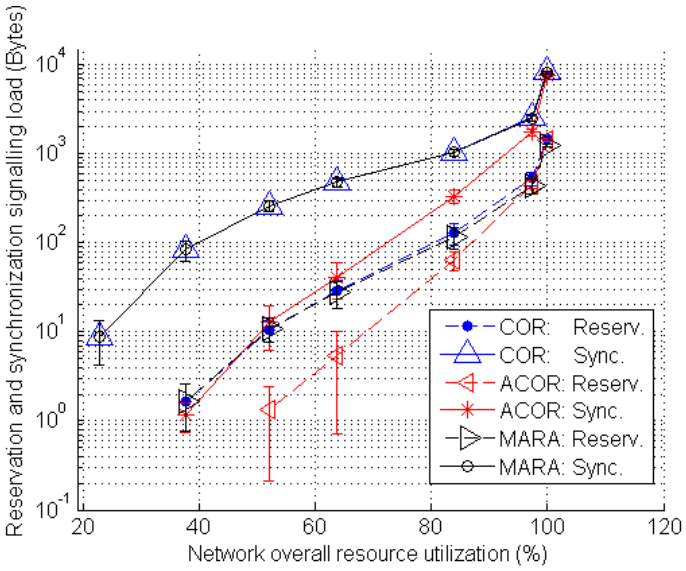


Figure 4.15. Number of reservation and synchronization signalling events.

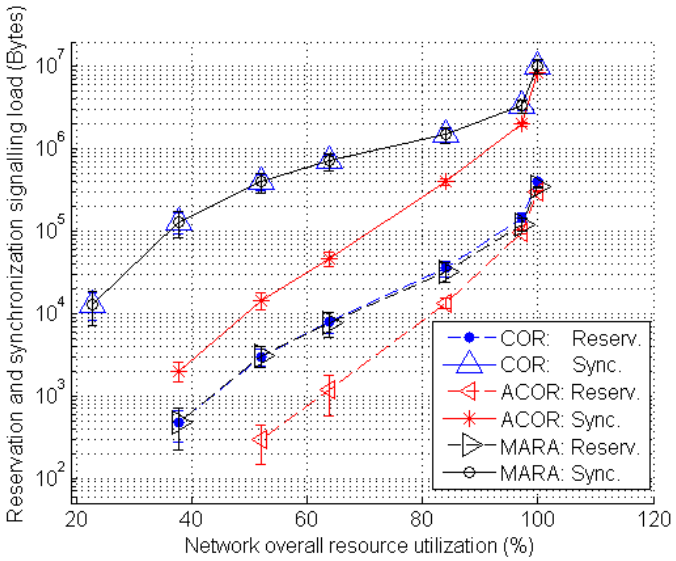


Figure 4.16. Reservation and synchronization signalling load.

Figure 4.17 shows that ACOR is able to reduce the overall signalling events of COR and MARA of a percentage ranging between 9% and 100%, and the signalling messages load of a percentage ranging between 15% and 100%, depending on the network resource utilization level. The percentage of the load reduction is higher than that of the events reduction in general. As we stated earlier in this sub-section, the relation between number of events and load is not linear, as it depends on the amount of information carried by the control messages triggered upon each event.

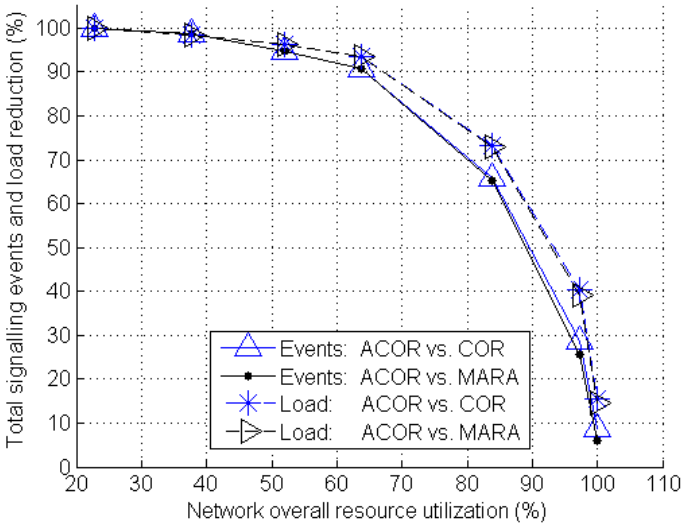


Figure 4.17. Total signalling events and load reduction of ACOR over COR and MARA.

Figure 4.18 shows the number of session requests that each of the algorithms has denied when there were still sufficient unused residual resources on the bottleneck outgoing interfaces of the candidate trees, as studied earlier analytically and seen in Figure 3.10. Hence, provided that the total unused residual resources are sufficient on the bottleneck outgoing interface of a requested

tree, it is confirmed that neither ACOR nor COR blocks incoming requests, and therefore, they effectively avoid wasting resources unnecessarily. Besides, MARA does not deny incoming requests unnecessarily when the network is under very low resource utilization conditions. However, MARA blocks many of the requests (more than 500 requests), especially with the start of the network congestion period of time. As it is described earlier, this is due to the inherent limitations of MARA's resource computation functions. Thus, it becomes apparent that ACOR is able to significantly reduce the overall control signalling overhead of the COR and the MARA without incurring unnecessary blockings or waste of resource, especially in dynamic scenarios with distributed network control decision points.

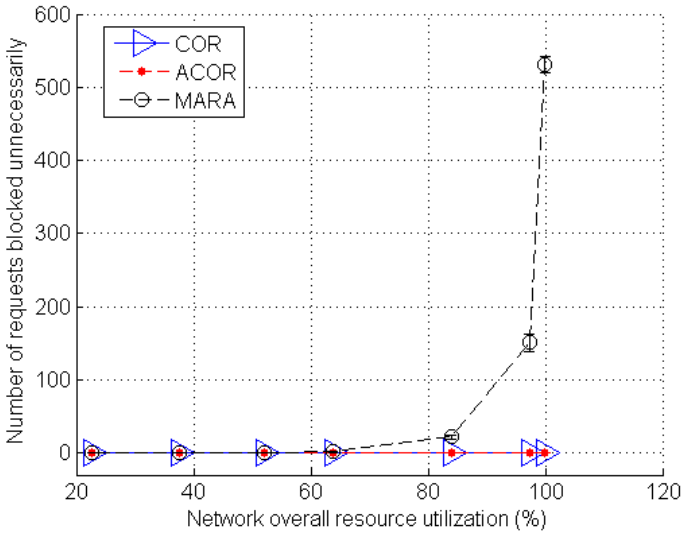


Figure 4.18. Number of denied requests while there were enough unused resources.

It is important to show that the ACOR control mechanism effectively supports differentiated QoS without QoS violation while implementing resource over-reservation dynamically in distributed network scenarios. Hence, Figure 4.19 is used to plot packet loss, while Figure 4.20 shows the delay experienced in each service CoS.

Moreover, the support of ACOR to effectively avoid QoS violation in distributed and dynamic scenario has also been studied through packets-based (with real traffics activated) simulation results. In particular, each network link capacity is set to 10Mbps and the bandwidth demand per request ranges between 128Kbps and 1Mbps. Then, the BE traffic sources are configured to generate packets at higher rate (out-of-profile) than expected. Besides, each traffic that belongs to EF CoS or to AF CoS is configured to comply with the rate granted to it (in-profile). Thus, we observe that only the BE traffic has experienced packets loss, between 0% and 18%, which was punished for the non compliance (out-of-profile). This demonstrates that misbehaviors of BE traffics did not affect other CoSs in terms of the packets dropping. However, one may argue that,

even though the BE traffic is out-of-profile, there are no dropping packets when the network overall utilization level is around 15%. Note therefore that WFQ is a work-conserving scheduler, which means that a network link is never empty as long as there is packet in a queue on the related outgoing interface.

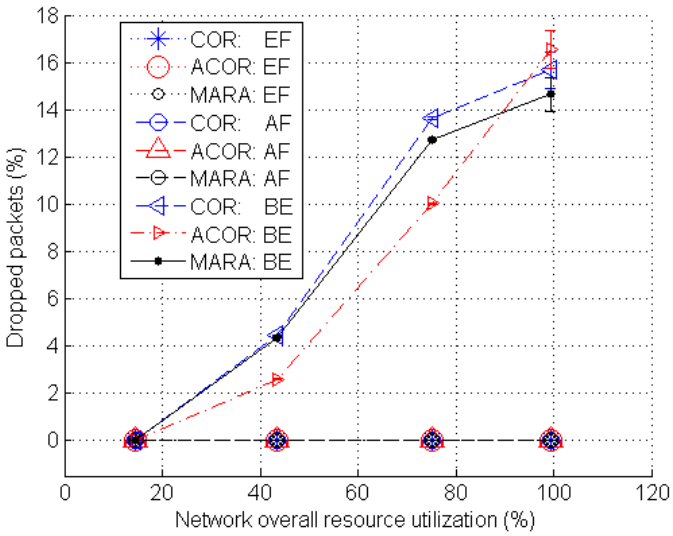


Figure 4.19. Packets loss with EF/AF traffic in-profile and BE traffic out-of-profile.

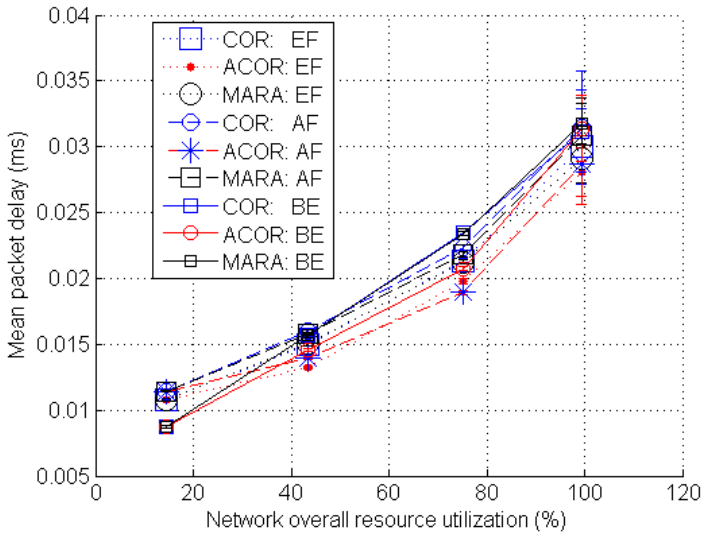


Figure 4.20. Packets delay with EF/AF traffic in-profile and BE traffic out-of-profile.

Moreover, Figure 4.20 provides the delay experienced by packets in each CoS. The delay increases slightly in all CoSs with the increase of network overall utilization level. Hence, at low link utilization level, the delay is very small. Traffic may experience less delay than they required, due to work-conserving scheduling discipline. More importantly and representative in Figure 4.20, we observe a steadier differentiated delay among the three CoSs when the network overall utilization level is around 100% (congestion time). In particular, the EF traffic flows experience the

lowest delay, the AF traffic shows longer delay, while the BE traffic incurs the longest delay. Thus, these packets based results confirm that the ACOR system effectively provide support for differentiation of services in dynamic scenarios.

4.4.3 **Discussion**

The main objective of this subsection is to provide a general discussion of our findings on aggregate resource over-provisioning based on the results obtained in this work. The analytical results in Figure 4.11 showed that more reservation surplus can effectively allow for more reduction of the signalling overhead since a single reservation signalling would leverage admission of several session requests. Further, Figure 4.12 depicted that, over-provisioning is even more attractive in terms of signalling overhead reduction in scenarios where sessions are mostly short-lived (with short lifetime). This is due to the fact that, the resources released by a session may be used by future one(s) without requiring a new reservation signalling. We also observed in Figure 3.10 that efficient over-reservation mechanism can avoid wasting resources. To achieve this, an over-reservation solution strongly requires a real-time knowledge of network topology and related links resources statistics to avoid QoS violations. Moreover, such view of network resources is of paramount importance to allow for efficient redistribution of residual reservations among various CoSs to prevent CoS starvations or waste of resources.

In the face of these challenges, existing solutions (e.g., BGRP, SICAP, MARA and COR) acquire network resource statistics periodically or on-demand basis using probing techniques and there is no synchronization among the edge nodes. As a consequence, they suffer from waste of resources which translates into unnecessary increase of session requests blocking probability. In order to minimize this performance issue, they prevent over-reserving too much resource and thus, fail to allow for the optimization of the signalling overhead reduction. In contrast to previous solutions, ACOR enables edge nodes to properly cooperate to obtain resource statistics in each CoS on each interface in a network in real-time manner under low signalling overhead. This way, ACOR allows for over-reserving as much resources as possible so that the reduction of signalling overhead can be optimized. Further, the analytical results in Figure 4.13 showed that, the increase of the number of communication paths on bottleneck links increases the signalling frequency due to the corresponding high demand of resources. Hence, one may improve performance by controlling (e.g., filtering) the link sharing of the paths inside a network, which is out of the focus of this Thesis.

In general, the ACOR demonstrates superiority over existing solutions by significantly reducing the signalling overhead without increasing session blocking probability unnecessarily, that is, without wasting resources. These results were validated through simulations carried out in ns-2,

which has been properly extended to support the ACOR functionalities. Hence, the reduction of signalling overhead is confirmed in Figure 4.15, Figure 4.16 and Figure 4.17, while the avoidance of resources wastage is shown in Figure 4.18. Moreover, the results in Figure 4.19 and Figure 4.20 are used to demonstrate the ACOR's support for differentiated QoS control which is of paramount importance for network and service convergence in the current and future class-based networks.

4.5 *Conclusion*

This chapter presented a novel multicast-based decentralization control model characterized by a well coordinated two-layer control mechanism to achieve improved performance. On one hand, ACOR implements advanced techniques for dynamic control of aggregate bandwidth over-reservation in networks deploying multiple distributed edge nodes without QoS violation, CoS starvation, waste of resources or unnecessary increase of service blocking, while keeping significantly low QoS reservation signalling overhead. Traffic flows are controlled and mapped to explicit edge-to-edge trees, and control load is pushed to the network border.

On the other hand, it uses a VOPR concept which consists in a way of virtually allocating the over-reservations of CoSs to each correlated communication path/tree on each link in the network. Thus, it is possible to process multiple services requests in each path/tree dynamically without being negatively affected by cross-traffics from other correlated paths as long as the VOPRs are available. Hence, the VOPR concept allows for enabling collaborating CDPs to require synchronization signalling only when there is VOPR exhaustion. Further, the cooperation is selective, that is, only the CDPs which are correlated with the information to be updated are dynamically included in the collaboration group, while unnecessary information exchange is also avoided and synchronization signalling overhead is also reduced.

As a result, ACOR is able to reduce both reservation and synchronization signalling and related overhead to improve control scalability. Moreover, the ACOR good knowledge of network topology and the related resource statistics acquired on real-time basis, keeping low signalling overhead is of paramount importance for key netwotk control sub-systems (e.g., admission control to avoid QoS violation, traffic engineering, etc.). We believe that this is a strong approach for QoS control in current and future class-based networks. One limitation of ACOR resides in the fact that it triggers synchronization whenever a VOPR exhausts upon receiving a session request. However, the VOPR would exhaust frequently or even on per request basis during network congestion period of time. This implies that ACOR would place per-flow synchronization at congestion time. Moreover, the increase of link sharing factor increases rapidly the synchronization frequency due to the granularity of the VOPR allocation per tree. Therefore, further investigation is still necessary to allow for optimizing the synchronization signalling overhead to effectively scale.

Chapter 5

Extended ACOR

Chapter 4 introduces a novel decentralization control mechanism, the ACOR, which uses aggregate bandwidth over-reservation and is able to provide good knowledge of network topology and related links resources statistics. In particular, ACOR enables distributed edge nodes configured as network CDPs to cooperate, exchange appropriate control information to synchronize to changes of resource states inside a network, so as to take control decisions with accurate information and increased resource utilization keeping low signalling overhead. A major approach in ACOR is its two-layering aggregate resource control mechanism, which consists in deploying resource over-reservation techniques such as ECOR (described in Chapter 3) to reduce QoS reservations overhead, and using VOPR to keep low synchronization signalling overhead.

While ACOR effectively allows for optimizing QoS reservations signalling and therefore the related processing overhead through ECOR, its performance of the VOPR allocation per multicast tree is limited by trees' density on interfaces, that is, the number of trees that share outgoing interfaces inside a network. In other words, the increase of the number of trees on outgoing interfaces, especially on bottleneck interfaces, rapidly increases the rate of the synchronization signalling between the collaborating CDPs. This is shown analytically in Figure 4.13 in Chapter 4, and thus raises scalability issues. Moreover, ACOR triggers synchronization signalling whenever the VOPR of a requested tree for a CoS is exhausted. This is very important to avoid VOPR starvation and waste of resources that increase session blocking probability unnecessarily. However, in critical network situations, as being close to congestion or congested, the VOPR may exhaust upon every session request due to resource unavailability. In these scenarios, ACOR is forced to synchronize per request, which would jeopardize performance, especially in large scale

networks or during prolonged network congestion periods of time. Therefore, further studies were still deemed necessary.

This chapter proposes the E-ACOR, a new approach that extends the ACOR architecture to manage VOPR aggregately per CDP, and not per tree, in the sense to alleviate performance dependence of ACOR from trees' density on network links. In addition, E-ACOR introduces a mechanism to efficiently track congestion information throughout a network in a way that allows for preventing unnecessary synchronization signalling when network or trees are congested. This way, E-ACOR aims to allow for reducing synchronization signalling rate, and to keep all the benefits of ACOR by assuring differentiated QoS under low QoS reservation signalling load without incurring QoS violations, unnecessary waste of resources or increase of service blocking probability. As such, the network overall performance can be optimized. Analytical and simulation results demonstrate the effectiveness of E-ACOR and its superiority over ACOR in terms of signalling overhead minimization and waste of resources, while guaranteeing improved QoS in dynamic and distributed networks.

This chapter is organized as follows. Section 5.1 describes the E-ACOR decentralization mechanism with focus on the main extensions to ACOR in terms of functionalities. Section 5.2 provides an analytical model of E-ACOR and section 5.3 discusses the performance evaluation. Finally, section 5.4 concludes the chapter.

5.1 *E-ACOR Control Mechanism*

The E-ACOR aims at optimizing the impact in the network performance of ACOR. For this purpose, E-ACOR introduces an *Aggregate VOPR* concept and a Congestion Information Tracking System (CITS), as two main features to improve the functionalities and behaviour of the CDPs. On one hand, E-ACOR proposes to aggregate VOPRs per CDP, in contrast to the fine-grained control in ACOR where the VOPR in a CoS on an outgoing interface is allocated per tree that uses the interface. The main objective of the VOPRs aggregation is to enable all trees rooted at the same CDP to share a common VOPR allocated to the CDP on their correlated outgoing interfaces, to allow for further reduction of synchronization rate since session demands are mostly unpredictable in individual trees. On the other hand, E-ACOR introduces the CITS mechanism, which enables each CDP to dynamically track congestion information about trees inside a network on real-time basis with low signalling and processing overhead. Hence, every CDP exploits appropriate congestion information of trees in a way that allows for preventing synchronization signalling when candidate trees are congested and incoming session requests would not be successfully admitted after synchronization.

In order to achieve these objectives, E-ACOR mainly improves the functionalities and interactions between components in ACOR, keeping in mind to assure compatibility with the latter. For example, we illustrate, using Figure 5.1, that E-ACOR maintains the general architecture of ACOR in which a system initialization phase may be characterized by multicast trees creation in *step (a)*, synchronization between CDPs in *step (b)*, and CDPs' local databases creation in *step (c)*, following the principles detailed in Chapter 4. In particular, E-ACOR extends the NetCIB functions in support for the information required by the Aggregate VOPR and the CITS mechanism, the Extended NetCIB (ENetCIB). Moreover, the AC functions of ACOR are extended to efficiently exploit the congestion information available in the ENetCIB to effectively avoid triggering unnecessary synchronizations. Besides, the SC functions are improved to prevent the CITS from placing undue signalling overhead by opportunistically using the VOPR synchronization messages for tracking congestion information. Notice that the RC functions of ACOR are reused, and the ECOR is embedded to optimize the reduction of QoS reservation signalling overhead. Moreover, E-ACOR reuses the BF functions implemented as ACOR-C agent (see Figure 5.1) which performs the basic functionalities required in all nodes. Thus, E-ACOR implements the BF, the RC and the extended AC, SC and NetCIB functions in software agent called E-ACOR-Edge (E-ACOR-E) agent to configure CDPs.

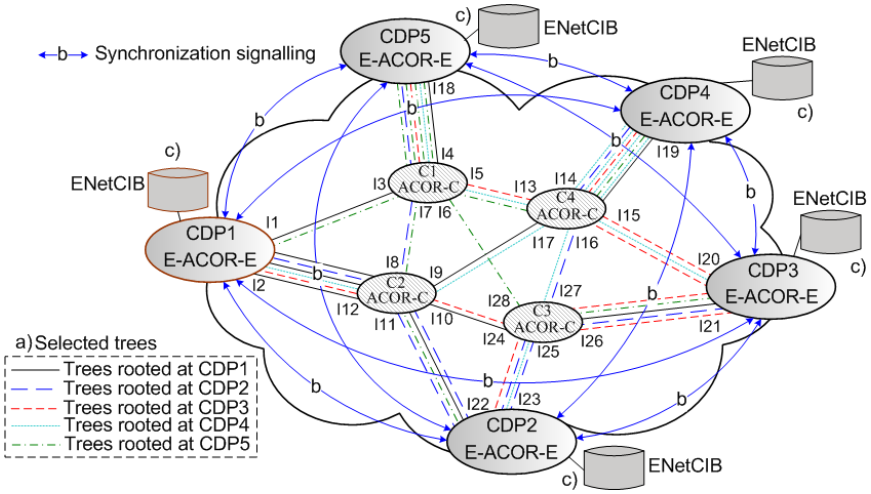


Figure 5.1. Illustration of E-ACOR decentralization network.

To ease the understanding, the remaining of this chapter will describe E-ACOR control mechanism with focus on the main extensions to the functionalities of ACOR and their technical effects on the performance.

5.1.1 Extension to VOPR Concept

In order to aggregate VOPRs per CDP, E-ACOR redefines the *Sharing Factor* of an outgoing interface, which denotes the number of ingress CDPs that deploy trees through the interface,

instead of the number of trees using the interface as in ACOR. The *Sharing Factor* $Factor_{CDP}(I_e)$ of any interface I_e in E-ACOR is thus limited to the number of correlated CDPs on the interface I_e. In order to facilitate understanding based on Figure 5.1, one can observe that the outgoing interface I_2 belongs to 3 trees, all of them with the same root CDP1, and each with leaf nodes CDP4 (tree 1), CDP3 (tree 2) and CDP2 (tree 3). In this case, the sharing factor of I_2 in E-ACOR ($Factor_{CDP}(I_e)$) is set to 1, whilst ACOR sets to 3. Thus, E-ACOR prevents the granularity of VOPR per tree, meaning that all the trees that a CDP deploys through an outgoing interface share the same pool of VOPR allocated to the CDP on the interface. The aggregate VOPRs can also be implemented as being the sum of the individual VOPRs of the trees which originate at the same CDP. Therefore, the VOPR aggregation allows for further reducing the frequency of VOPR exhaustion or synchronization signalling events where the VOPR per tree would exhaust more often due to highly unpredictable service demands to each tree inside a network. The aggregate VOPR of a CDP_A in a CoS_i on an outgoing interface I_e, denoted as $Vopr(i, I_e, CDP_A)$, is therefore a share of the bandwidth over-reservation of the CoS_i for the CDP_A on the interface I_e, and is obtained as:

$$Vopr(i, I_e, CDP_A) = U_{Agg}(i, I_e, CDP_A) + \frac{R_{BW}(i, I_e) - U_{BW}(i, I_e)}{Factor_{CDP_A}(I_e)} \quad (5.1)$$

where $R_{BW}(i, I_e)$ and $U_{BW}(i, I_e)$ are the reservation and used bandwidth obtained as in equation (4.1), and $U_{Agg}(i, I_e, CDP_A)$ is the total amount of bandwidth granted to the currently active flows (aggregate used bandwidth) in CoS_i on all the trees T_x deployed by the CDP_A through the outgoing interface I_e, and is obtained using the equation (4.3) where m is the number of the trees of the CDP_A.

5.1.2 Extension to NetCIB Database

The E-ACOR architecture embeds the database called the ENetCIB. In particular, the E-ACOR's ENetCIB extends the TOPOLOGY and VOPRs tables of ACOR's NetCIB, and introduces a new table called CONGESTION, to store congestion information about existing trees as in the following.

TOPOLOGY Table: the TOPOLOGY table stores the interface Sharing Factor of each outgoing interface as being the number of correlated CDPs on the interface (see subsection 5.1.1). Moreover, it stores the CDP's Aggregate Used bandwidth obtained using equation (4.3) for each CoS on each outgoing interface of the CDP's trees.

VOPRS Table: the VOPR table of a CDP_A in E-ACOR stores the aggregate VOPRs of the CDP_A in each CoS_i on each outgoing interface I_e that lies on the CDP_A's trees as defined by equation (5.1). A CDP's tree is a tree that originates at the CDP.

CONGESTION Table: this (Table 5.1) is introduced to track congestion information about trees inside a network. The main idea is to enable every CDP to exploit appropriate congestion information in a way that can prevent unnecessary synchronization signalling when all candidate trees are congested and incoming requests cannot be admitted after synchronization. To facilitate the understanding, our description is illustrated with the CONGESTION table of CDP1 from Figure 5.1.

Table 5.1. CONGESTION Table.

Tree Index	CDP1		CDP2	CDP3	CDP4	CDP5
	Availaible Bandwidth	*Congestion Flag*	*Congestion Flag*	*Congestion Flag*	*Congestion Flag*	*Congestion Flag*
0	999.5	0	0	0	0	0
1	999.5	0	0	0	0	0
2	999.5	0	0	0	0	0
3	999.5	0	0	0	0	0

As shown in Table 5.1, each CDP maintains the ID of the trees inside a network, where the ID of a CDP's tree is a tuple composed of the CDP's ID and the tree's index *(CDP_ID, Tree_Index)* to assure uniqueness. Note therefore that the indices of trees rooted at different CDPs may overlap, since they are assigned by CDPs independently. In Table 5.1, the indices of CDP1's trees ranges from 0 to 3 as in Figure 5.1, since it only maintains a single tree from itself to each of the remaining 4 CDPs for the sake of simplicity in our illustration. Further, each CDP (e.g., CDP1 in Table 5.1) maintains the total available bandwidth $A_{BW}(I_b, T_x)$ on the bottleneck outgoing interface I_b of each of its own trees, using the following function:

$$A_{BW}(I_b, T_x) = \min\left\{C - b - \sum_{i}^{k} U_{BW}(i, I_e)\right\} \tag{5.2}$$

where, C is the capacity of each outgoing interface I_e on its own tree T_x, b is the bandwidth dedicated to the control CoS, and $U_{BW}(i, I_e)$ is the total amount of the used bandwidth in each service CoS_i on the interface I_e; these parameters are obtained from the TOPOLOGY table (Table 4.2 in Chapter 4). This is illustrated in Table 5.1, considering that each outgoing interface in Figure 5.1 has a capacity (*C=1000Mbps*), a certain amount of bandwidth (*b=0.5Mbps*) is dedicated to the Control CoS on each interface, and the total used bandwidth is null as at network initialization.

The available bandwidth $A_{BW}(I_b, T_x)$ parameter obtained in equation (5.2) enables each CDP (e.g., CDP1) to be aware of the congestion level of each of its trees. Unfortunately, this parameter

is not updated on real time basis; it is updated only upon synchronization since network bandwidth is shared by multiple trees originated at distributed CDPs under unpredictable traffic behaviour. This imposes that a CDP cannot simply rely on the available bandwidth shown in the CONGESTION table to avoid synchronization without taking wrong decisions or wasting resources. E-ACOR addresses this challenge by associating another control parameter, the *Congestion Flag*, to each of the trees inside a network. In particular, a Congestion Flag of a tree T_x is either ON *(Flag(T_x) = 1)* to refer that the related tree is set to congestion status, or it is OFF *(Flag(T_x) = 0)* to indicate that the related tree is not set to congestion status. Hence, all Congestion Flags are initialized to OFF. The way the Congestion Flags of trees are jointly exploited with the total available bandwidth of trees to allow for preventing unnecessary synchronization signalling without wasting resources is detailed in the rest of this chapter.

5.1.3 Extension to Admission Control Functions

When a network is running and a given ingress CDP_A receives an authorized service request r_i to a CoS_i and destined to a given egress CDP_B in the control domain, CDP_A collects the candidate trees of the incoming request, that is, the trees it roots to connect the desired egress CDP_B. Then, among the candidate trees, it selects the one (T_x) holding the highest available VOPR. An available VOPR in a CoS_i on a CDP_A's tree T_x, denoted $A_{Vopr}(i,T_x)$, is the aggregate VOPR of the CDP_A which has not yet been granted to any flow in CoS_i on the bottleneck outgoing interface of the tree T_x. It is obtained by the following function:

$$A_{Vopr}(i,T_x) = \min\left\{Vopr(i,I_e,CDP_A) - U_{Agg}(i,I_e,CDP_A)\right\} \tag{5.3}$$

where $Vopr(i,I_e,CDP_A)$ is obtained using the equation (5.1), $U_{Agg}(i,I_e,CDP_A)$ is obtained using the equation (4.3), and I_e is an outgoing interface on the tree T_x.

If the admission is successful ($r_i \le A_{Vopr}(i,T_x)$), CDP_A maps the request to the requested CoS_i on that tree T_x without synchronization or QoS reservation signalling event. Then, the CDP automatically updates its used bandwidth in the TREES table, as well as the *Aggregate Used* bandwidth in the admitted CoS_i for each of the outgoing interfaces of the tree T_x in its TOPOLOGY table, according to the bandwidth r_i granted to the flow. This way, every CDP updates its Aggregate Used bandwidth in all relevant CoSs and interfaces on real-time basis in its local database. As a result, the available VOPRs are also deduced in real-time manner using equation (5.3), and thus allowing each CDP to admit several service requests without synchronization or QoS reservation signalling into the network, as long as the VOPRs are available. Likewise,

whenever a service belonging to a CoS_i terminates from a tree T_x, the CDP automatically updates its used bandwidth in the TREES table and the *Aggregate Used* bandwidth in the concerned CoS_i for each of the outgoing interfaces of the tree T_x in its TOPOLOGY table accordingly. Note that, there is no QoS release signalling into a tree from which a service terminates, since the bandwidth is over-reserved.

However, if the attempt to admit a request based on available VOPR fails ($r_i > A_{Vopr}(i,T_x)$), being the unused VOPR insufficient on the bottleneck outgoing interfaces of all candidate trees, let's recall that ACOR immediately triggers synchronization among the correlated CDPs of the candidate trees in order to avoid QoS violations or waste of resources. While synchronization is imposed by unpredictable service demands and the resource sharing by distributed CDPs, it forces ACOR to unnecessary synchronization signalling per session request due to the VOPRs exhaustions during congestion period. E-ACOR addresses this challenge by introducing the CITS, which is able to make use of trees' congestion information dynamically to avoid unnecessary synchronization signalling without incurring waste of resources or blocking services unnecessarily. This mechanism is detailed in the subsequent subsection.

5.1.4 **Extension to Synchronization Control Functions**

The E-ACOR congestion information tracking mechanism maintains trees' congestion information on real-time basis using Table 5.1, which enables each CDP to avoid unnecessary synchronization signalling during congestion time. As we detailed in subsection 5.1.2, a tree's congestion information includes its total available bandwidth and the associated congestion Flag. Hence, this subsection focuses on how this information is used to achieve performance.

To facilitate the understanding, we consider that the VOPR in a CoS_i on a tree T_x exhausts ($r_i > A_{Vopr}(i,T_x)$), and the available bandwidth seen in CONGESTION table (please see Table 5.1) is insufficient to admit an incoming request ($r_i > A_{BW}(I_b,T_x)$). In this case, one may consider that the tree T_x is congested, and block the incoming request without synchronization signalling. Unfortunately, the available bandwidth $A_{BW}(I_b,T_x)$ is updated only upon synchronization as we explained earlier in subsection 5.1.2. In other words, note that certain traffic flows may terminate from correlated trees of a T_x, and the available bandwidth in the tree T_x may increase in the meantime between two synchronization events. Therefore, denying session requests based on previous knowledge of available bandwidth $A_{BW}(I_b,T_x)$ only without synchronization would increase session blocking probability unnecessarily or waste resources. A simple solution to cope with this issue may consist in enabling every CDP to advertise the correlated CDPs (with the

amount of bandwidth released and the related tree's ID) whenever there is a service termination, so that each CDP can update the related available bandwidth on real-time basis. However, while this would allow for avoiding synchronization signalling, it would lead to excessive release notification signalling and database processing overhead between correlated CDPs.

In this sense, E-ACOR proposes the joint use of available bandwidth parameters and congestion *Flags* associated to trees to achieve scalable performance without wasting resources as further detailed in the following steps.

- ***Step 1:*** *VOPR exhausted ($r_i > A_{Vopr}(i,T_x)$) while available bandwidth is insufficient in all candidate trees ($r_i > A_{BW}(I_b,T_x)$) upon receiving a session request r_i, and the congestion Flag of the tree T_x which shows the highest available resource is OFF ($Flag(T_x)=0$).*

  In this case, the CDP turns the congestion Flag ON ($Flag(T_x) = 1$) and triggers synchronization, including the updated Flag status of T_x in the synchronization message for the remote CDPs to update the Flag status of T_x accordingly. Thus, each relevant CDP is enabled to update the status of T_x's flag accordingly without any extra signalling message.

  Hence, the control metrics, such as the total used bandwidth, the VOPRs and the reservations parameters of the outgoing interfaces of certain candidate trees, may be readjusted to allow for session admission. In case the incoming service request is successfully admitted, the congestion Flag of the tree T_x is set to OFF and included in the synchronization message sent to correlated CDPs after the local database is thus processed. However, if the incoming request is not admitted after the synchronization, the status of T_x's Flag is left ON.

- ***Step 2:*** *VOPR exhausted ($r_i > A_{Vopr}(i,T_x)$) while available bandwidth is insufficient in all candidate trees ($r_i > A_{BW}(I_b,T_x)$) upon receiving a session request r_i, and the congestion Flag of the tree T_x which shows the highest available resource is ON ($Flag(T_x)=1$).*

  In this case, the CDP considers that the candidate trees are congested and blocks the incoming request without synchronization signalling. The request may be mapped to a CoS with lower QoS level depending on the service agreement between customer and service provider.

- ***Step 3:*** *VOPR exhausted ($r_i > A_{Vopr}(i,T_x)$) and available resource is sufficient in a candidate tree T_x ($r_i \leq A_{BW}(I_b,T_x)$) upon receiving a session request r_i.*

In this case, the CDP automatically triggers synchronization to allow for updating the resource status on all relevant interfaces, so that relevant metrics such as the total used bandwidth, the VOPRs and the reservations parameters, may be readjusted upon need for the admission purpose. In this situation, the congestion Flag of the tree T_x is turned OFF if it was ON, and is included in the synchronization message for the remote CDPs to update the Flags status accordingly.

- ***Step 4:*** *Procedure upon session termination from a tree T_x.*

  In this case, the CDP takes the following actions. In case the Flags of the correlated trees of T_x happen to be ON, the CDP turns them OFF and sends a notification message to the corresponding CDPs carrying the ID of the tree T_x from which the session terminated. This way, the concerned CDPs are able to reset their Flags accordingly to allow for future synchronization upon need, for the purpose of reusing the resources made available by the terminated session. However, if none of the Flags of the correlated trees of T_x is ON, the CDP does not issue any notification. Also, a CDP resets the Flag of its tree T_x upon service termination from the tree without issuing notification to the correlated CDPs. This way, E-ACOR aims to allow for preventing unnecessary synchronization without being forced to notify correlated CDPs upon every service termination, and scalability can be achieved in a way that avoids blocking service requests unnecessarily or wasting resources. Moreover, no service is admitted without assuring a minimum available reservation, and thus avoiding QoS violations.

5.2 *E-ACOR Analytical Model*

This section provides a model to compare the performance of the E-ACOR aggregate VOPR with that of the ACOR VOPR described in Chapter 4 in terms of minimization of synchronization signalling frequency. For this purpose, we use the network topology in Figure 5.2 to illustrate bottleneck link sharing scenarios where each ingress CDP deploys several trees from itself to each of the egress CDPs.

Then, we consider that a certain amount of bandwidth $R_{BW}(i, I_b)$ is over-reserved to each CoS_i on a bottleneck outgoing interface I_b inside the network. Besides, service requests arrival to a CoS_i on a given tree T_x that uses the interface I_b is as Poisson process with rate $\lambda_{i,x}$, and therefore, the sum λ_i of service requests arrival rates to a set of trees in a CoS_i on the I_b is also a Poisson process. In this sense, the overall session requests arrival rate $\lambda_{i,CDPx}$ from all *m'* trees of a CDP_x into the CoS_i over the interface (**A→ B**) is also as Poisson process such that:

$$\lambda_{i,CDPx} = \sum_{x=1}^{m'} \lambda_{i,x} \tag{5.4}$$

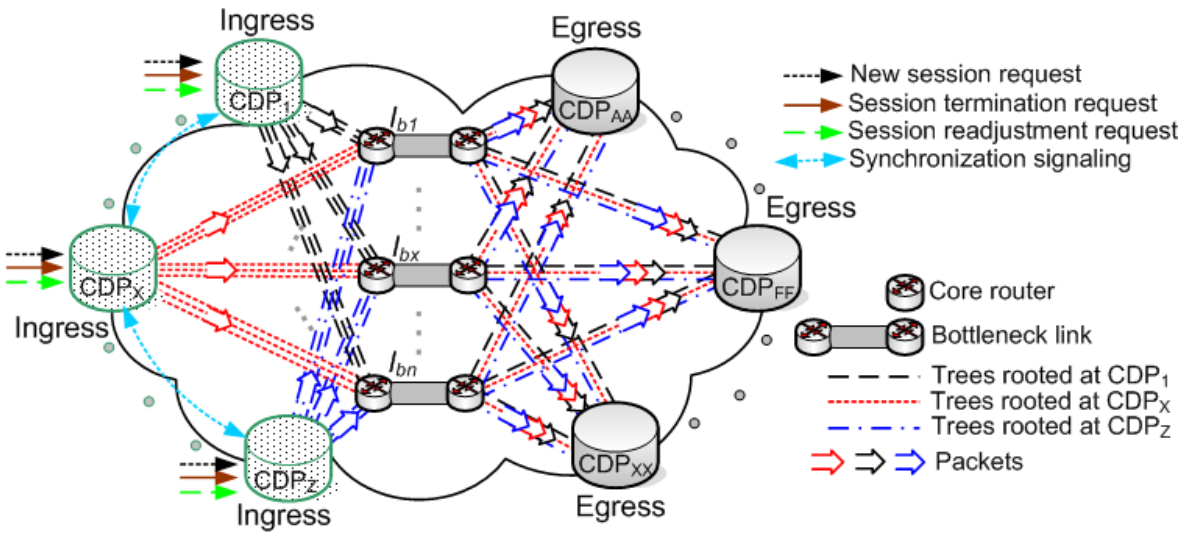


Figure 5.2. Topology for analytical study.

Hence, considering that m number of trees rooted at q different ingress CDPs share the bottleneck interface I_b, we model the performance of the aggregate VOPR and that of the ACOR VOPR as in the following.

ACOR: the available VOPR $A_{Vopr}(i,T_x)$ of a given tree T_x in a CoS_i within a surplus of reservation $R_{BW}(i,I_b)$ on the interface I_b is obtained as in equation (4.9), and the total number n of sessions that it can accommodate simultaneously without requiring synchronization event for a given tree T_x sharing the interface I_b is obtained using equation (4.10).

E-ACOR: the available aggregate VOPR $A_{AggVopr}(i,CDP_x)$ of a given CDP_x in a CoS_i within $R_{BW}(i,I_b)$ on the interface I_b is obtained by:

$$A_{AggVopr}(i,CDP_x) = \frac{R_{BW}(i,I_b)}{q} \tag{5.5}$$

Then, the total number n' of sessions that an available aggregate VOPR $A_{AggVopr}(i,CDP_x)$ of CDP_x can accommodate simultaneously in a CoS_i without requiring synchronization event for any of its correlated tree T_x on the interface I_b is expressed as:

$$n' = \left\lfloor \frac{R_{BW}(i,I_b)}{r_i * q} \right\rfloor \tag{5.6}$$

This way, we use the analytical study principles detailed in section 4.3 in Chapter 4, and model the ACOR VOPR as an M/M/n/n queuing system and the aggregate VOPR of E-ACOR as an M/M/n'/n' queuing system. As such, the probability $P_{i,x}^{ACOR}$ that the available VOPR $A_{Vopr}(i,T_x)$ (see equation (4.9)) exhausts to trigger synchronization in ACOR is obtained as in equation (4.11). Likewise, the probability $P_{i,x}^{A2COR}$ that the available aggregate VOPR $A_{AggVopr}(i,CDP_x)$ (see equation (5.5)) exhausts to trigger synchronization in E-ACOR systems is the probability that an incoming request to the CoS_i on any of the CDP_x's tree on the bottleneck interface finds all the *n'* "VOPRed sessions" slots occupied, and is also obtained using Erlang B formula as:

$$P_{i,x}^{A2COR} = \frac{\left(\frac{\lambda_{i,CDPx}}{\mu}\right)^{n'} * \frac{1}{n'!}}{\sum_{\alpha=0}^{n'}\left(\frac{\lambda_{i,CDPx}}{\mu}\right)^{\alpha} * \frac{1}{\alpha!}} \tag{5.7}$$

where μ is a real number (service rate) as in equation (4.11) and λ_{i,CDP_x} is obtained as in equation (5.4).

5.3 *Performance Evaluation*

In order to show the benefits of E-ACOR in comparison with ACOR, our performance evaluations are carried out as in the following. First, the advantages of VOPRs aggregation of E-ACOR over the per-tree VOPR approach of ACOR, as studied in section 5.2, are assessed analytically. Furthermore, we focus our analysis on the overall improvement of E-ACOR over ACOR in terms of minimization of synchronization signalling overhead through simulations using the ns-2 [233], which was appropriately extended with ACOR and E-ACOR functionalities. The performance characteristics concerning QoS reservation signalling overhead minimization, avoidance of QoS violations and waste of resources are incorporated in E-ACOR as inherent advantages of ACOR (see Chapter 4), and therefore, the related results are not repeated in this chapter. In order to clearly observe the advantages of VOPRs aggregation and the benefits of the congestion tracking mechanism of E-ACOR separately through simulations results, we implement a third control mechanism called C-ACOR. Basically, the C-ACOR is equal to ACOR embedding the congestion tracking mechanism of E-ACOR, but without the VOPR aggregation. In other words, the C-ACOR is a combination of VOPR per tree and the congestion tracking mechanism of E-ACOR. Then, we plot the simulation results for these three approaches (E-ACOR, ACOR and C-ACOR) in a way that facilitates analysis of each enhancement function (VOPR aggregation and congestion information tracking mechanism) of E-ACOR over ACOR.

5.3.1 **Analytical Parameters Configuration**

To facilitate the understanding of the description, the ingress CDPs which deploy trees through I_b are called *correlated ingresses*, and the egress CDPs which are connected through I_b are called *correlated egresses*. Moreover, each correlated ingress deploys at least 1 tree and at most N' trees through the interface I_b, such that an ingress deploys through I_b at most 1 tree to any of the egresses. This leads to a lower bound and an upper bound of *Sharing Factor* of the bottleneck interface I_b, called respectively the *Smallest Sharing Factor* (SSF) and the *Highest Sharing Factor* (HSF). Basically, SSF corresponds to a scenario where each correlated ingress deploys 1 tree through I_b, and HSF corresponds to a scenario in which each correlated ingress maintains N' trees through I_b. Hence, given a number *c_Ing* of correlated ingress CDPs and a number *c_Eg* of correlated egress CDPs for I_b, the corresponding lower and upper bounds of sharing factors are obtained as:

$$SSF(I_b) = \begin{cases} c\_Ing, & \text{in ACOR} \\ c\_Ing, & \text{in E-ACOR} \end{cases} \tag{5.8}$$

$$HSF(I_b) = \begin{cases} c\_Ing * c\_Eg, & \text{in ACOR} \\ c\_Ing, & \text{in E-ACOR} \end{cases} \tag{5.9}$$

Then, we use the SSF and the HSF to compare the lower and upper bounds of synchronization rate of E-ACOR and ACOR approaches by varying the number of ingress CDPs (similar when varying the number of egress CDPs) on one hand. On the other hand, we vary the amount of bandwidth over-reservation in CoS_i, and show results as in subsequent subsection.

5.3.2 **Analytical Results**

Figure 5.3 is used to plot the probability of synchronization signalling frequency in ACOR and in E-ACOR, as a function of the number of correlated ingresses. In particular, the number of correlated ingresses is varied between 1 and 20 with a fixed number of 12 correlated egresses. The configurations of the scenario are summarized in Table 5.2.

Table 5.2. Analytical parameters configurations.

$B_i = 100$	Over-reserved BW in CoS_i on the interface I_b (Mpbs)
$\bar{r} = 1$	Mean bandwidth requested per session (Mbps)
$\mu = 3$	Mean service rate per session (requests/time unit)
$\lambda_{i,x} = 10$	Requests arrival rate to a given tree T_x in CoS_i on I_b
$N = 20$	Total number of ingress CDPs
$N' = 12$	Total number of egress CDPs

Analytical results obtained using Erlang B formula in (4.3) in Chapter 4 and Figure 5.3 show that E-ACOR incurs lower synchronization signalling rate than ACOR, as we expected due to the aggregation of the VOPRs in E-ACOR. We also observe that the synchronization frequency grows more rapidly in ACOR than in E-ACOR with the increasing number of ingress CDPs; it reaches about 100% probability of signalling in ACOR when the number of ingress CDPs is beyond 9. This shows the limitation of VOPR granularity per tree in ACOR. E-ACOR and ACOR show the same lower bound synchronization rate. Note that in the lower bound scenario as we defined earlier, every correlated ingress CDP deploys only one tree through the bottleneck interface I_b, and thus showing no difference between the aggregate VOPR of E-ACOR and the VOPR per tree of ACOR to confirm results.

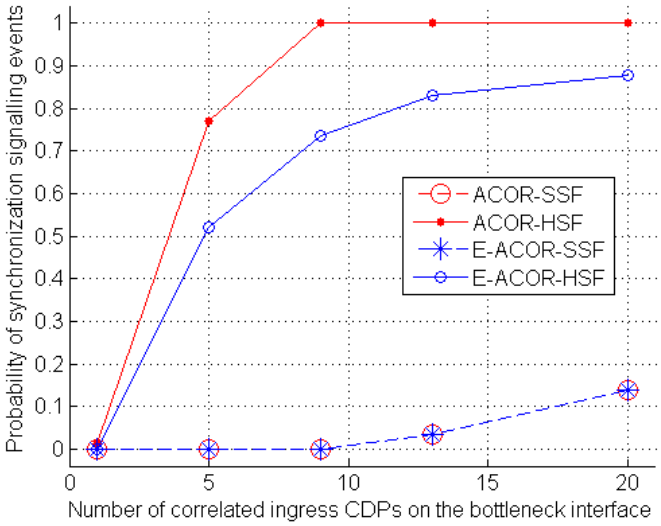


Figure 5.3. Effect of number of ingress CDPs on signalling events.

Figure 5.4 shows the probability of synchronization signalling event rate in ACOR and in E-ACOR as a function of the amount of bandwidth over-reserved for the CoS_i on the bottleneck outgoing interface I_b. The scenario is based on the parameter configurations in Table 5.2, with the only difference that the amount of over-reserved bandwidth $R_{BW}(i, I_b)$ to the CoS_i is varied between 50 Mbps and 2Gbps. E-ACOR shows better performance than the ACOR approach since the upper bound of synchronization frequency in E-ACOR is lower than that of ACOR. Figure 5.4 also confirms that E-ACOR and ACOR experience the same lower bound synchronization frequency as expected.

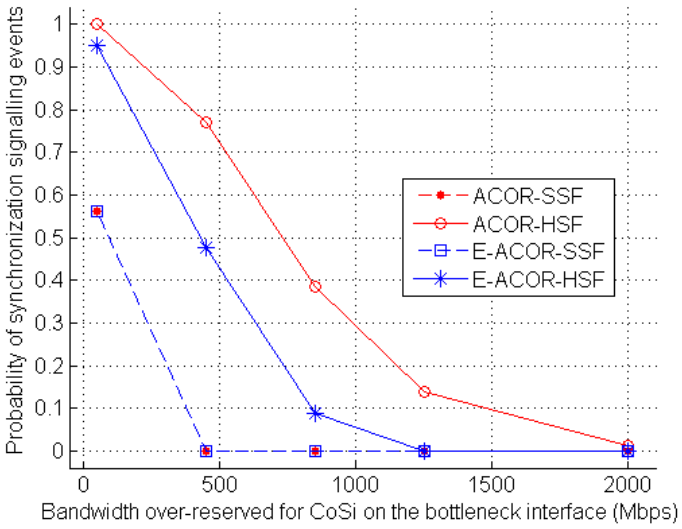


Figure 5.4. Effect of over-reserved bandwdith on signalling events.

5.3.3 **Simulation Scenario and results**

The simulation setup is carried out as described in subsection 4.4.2 in Chapter 4. In the methodology we adopted for the network overall resource utilization level, any of the 4 networks simulated gets congested with about 10,000 session requests. However, we performed each simulation with 25,000 requests such that allowing for observing prolonged network congestion periods of time to study the benefits of congestion tracking mechanism of E-ACOR more precisely. The simulation is run 5 times with different seeds of random mapping of requests to CoSs, CDPs and egress routers, for each topology. Then, the averaging values are plotted for all topologies and seeds with a confidence interval of 95%.

Then, we collected the number of synchronization signalling events and signalling load in E-ACOR, in ACOR, and in C-ACOR. We analyzed also the minimization of synchronization signalling number of E-ACOR in terms of percentage, to ease the comparison between the different approaches.

Figure 5.5 depicts the number of signalling events used for synchronization over the number of session requests in the experiments configured with ACOR, C-ACOR and E-ACOR solutions, and Figure 5.6 plots the corresponding signalling messages load. Recall that the load is obtained based on the ACOR-P described in section 4.2 in Chapter 4.

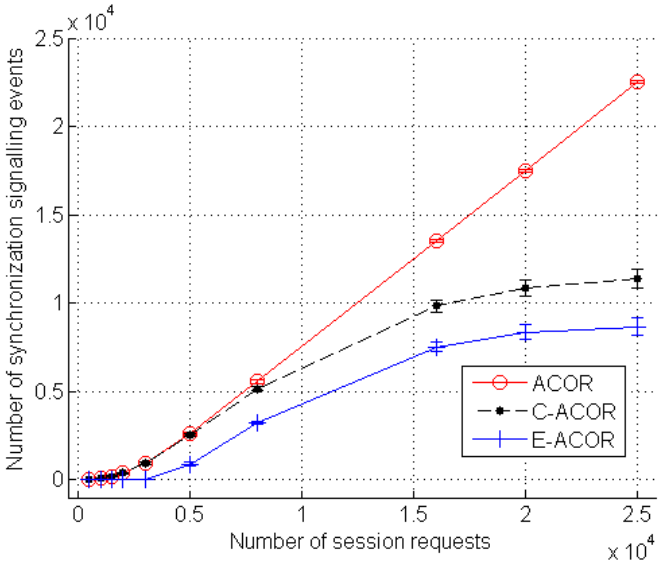


Figure 5.5. Number of synchronization signalling events.

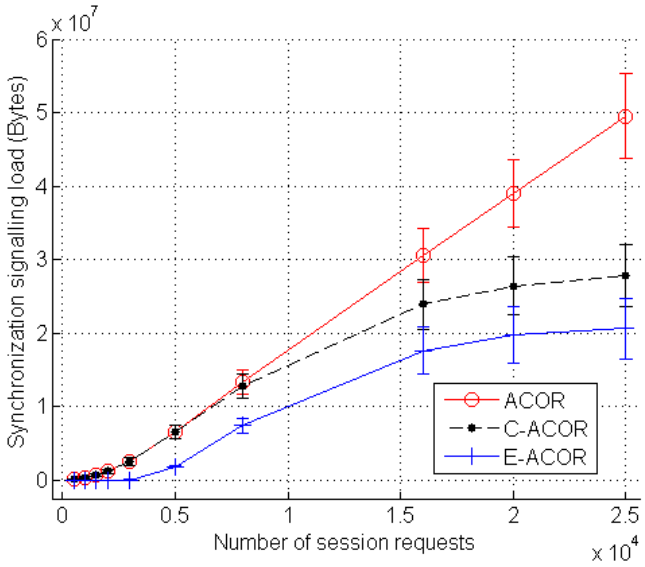


Figure 5.6. Synchronization signalling load.

The results of Figure 5.5 show that E-ACOR allows for significantly reducing the overall synchronization signalling event number of both the ACOR and the C-ACOR. Note that, in the following results, we count release notification signalling events of E-ACOR as synchronization event. To clearly observe how much signalling is reduced by means of VOPRs aggregation and how much is achieved through the congestion tracking mechanism of E-ACOR, let's first compare the C-ACOR with the ACOR. Recall that C-ACOR consists of ACOR integrated with the congestion tracking mechanism of E-ACOR. Hence, one can observe that, under low network resource utilization level with less than 5.000 requests, there is no noticeable difference between ACOR and the C-ACOR, as is expected since the communication trees may not have experienced congestion as yet. Indeed, C-ACOR effectively demonstrates increasing superiority over ACOR

with the increasing overall resource utilization level (beyond 5,000 requests). This implies that certain trees may happen to start getting congested when the number of requests is larger than 5.000.

When comparing E-ACOR with the C-ACOR, since both implement the same congestion tracking mechanism of E-ACOR, it becomes clear that E-ACOR outperforms the C-ACOR due to the VOPRs aggregation techniques, which makes E-ACOR less sensitive to trees density on bottleneck outgoing interfaces inside a network. Therefore, the E-ACOR is able to drastically reduce synchronization signalling further when compared with ACOR. To better appreciate the benefits of E-ACOR, Figure 5.7 is used to show its synchronization signalling events and load reduction in terms of percentage.

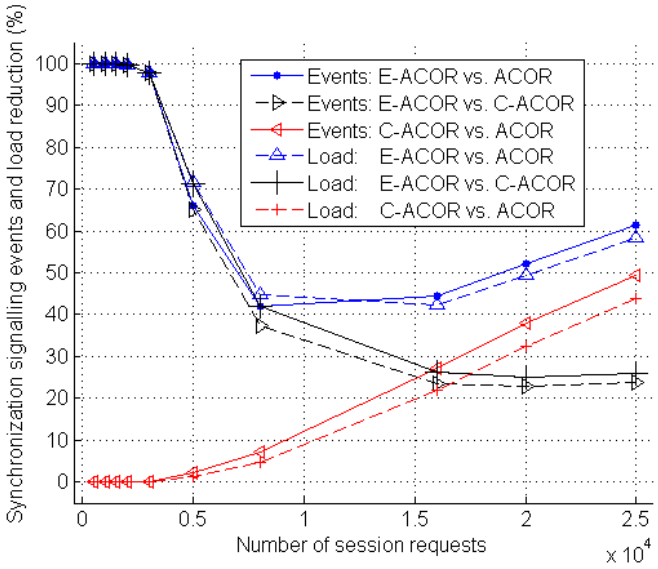


Figure 5.7. Reduction of synchronization signalling events number and load.

First, we compare the performance of C-ACOR with that of ACOR. Thus, we observe that, during very low network utilization level (below 2.000 requests), the C-ACOR stays equal to ACOR, since the trees may not have yet experienced congestion. However, the former outperforms the latter very rapidly as the network resource utilization increases (bottleneck outgoing interfaces are experiencing congestion). This shows that the congestion tracking mechanism is effectively useful to achieve scalability during network congestion periods of time.

Second, we compare the E-ACOR with the C-ACOR and observe that, during very low network utilization level (below 2,000 requests), the E-ACOR is able to reduce up to 100% of the synchronization events and load of the C-ACOR. This implies that E-ACOR does not generate signalling events in these conditions, where the trees may be far from being congested. One can see that the aggregation of VOPRs can profit more from resource sharing than the VOPR per tree, and thus, allows for further avoiding unnecessary synchronization signalling even at low resource

utilization level on the bottleneck interfaces. During prolonged network congestion periods of time, E-ACOR maintains its superiority above 22% over the C-ACOR. This shows that the VOPR aggregation concept is important to reduce the synchronization signalling overhead during network operations time.

E-ACOR allows for significantly reducing the synchronization signalling overhead of ACOR above 40%, since it combines both the VOPRs aggregation and the congestion tracking mechanism. Moreover, the signalling overhead (events and load) reduction increases rapidly during prolonged congestion period of time, which is of paramount importance to improve system overall performance. It also important to mention that, no service is admitted while there is not sufficient resource for admission and QoS violation is also avoided.

5.4 *Conclusion*

This chapter presented a novel approach, the E-ACOR, for resource and admission control, which is able to allow for optimizing control scalability in terms of signalling and related overhead reduction in class-based networks. In order to achieve this, E-ACOR implements appropriate techniques to significantly reduce QoS reservation signalling overhead with increased resource utilization, by aggregating bandwidth over-reservation control. Moreover, E-ACOR is able to track congestion information on bottleneck interfaces throughout a network in a way that enables self-controlling CDPs to avoid unnecessary synchronization signalling during bottleneck interfaces congestion periods of time.

We believe that this approach is of high importance to enhance performance in the networks. Therefore, the survivability control of ACOR, which also applies to E-ACOR, will be studied in Chapter 6 with the objective of finding ways to provide support for stable operations and service continuity in the presence of unpredictable links and nodes failures.

Chapter 6

Survivable ACOR Mechanism

The communication systems have become integral part of our society while unpredictable failures, usually caused by natural disasters (e.g., fire, earthquake, etc.), malicious attacks, hardware faults, and human mistakes, threaten their normal operations. The term survivability refers to the ability of a network to assure service continuity to a certain degree in the presence of these challenges. The survivability approaches are generally classified into protection-based and restoration-based [56], [55] with a main objective to achieve service stability through minimum recovery time, while assuring differentiated control and maintaining maximum resource utilization [57]. The protection-based techniques provide pre-defined backup paths, meaning that, at least a protection path is provided as soon as a working path is setup. A backup path may be dedicated or shared (e.g., 1+1 or 1:1 architectures) [194], such that traffic can be quickly switched over in case of failure on protected paths. In 1+1 architecture, identical traffic is transmitted simultaneously on both the working and protection entities, while in 1:1 architecture, the protection entity may be shared by low-priority traffic to increase resource utilization, since this traffic can be preempted in case of failure of the protected entity. The protection may be local to a link/node or global by bypassing an entire working path, and guarantees fast recovery within time frames amounting to tens of milliseconds. However, it is expensive since spare resources must be provisioned without a priori knowledge of failures' patterns. The restoration mechanisms are more cost-effective by establishing alternative paths only after a failure has occurred. Nonetheless, restoration actions may be completed within periods ranging from hundreds of milliseconds to a maximum of a few seconds due to the delay in finding appropriate resources after failures. Therefore, the protection and the restoration techniques can be combined in practice to improve performance [186].

This chapter proposes the SACOR that pushes survivability control load and complexity to CDPs for fast recovery and scalability purposes. It provides *VOPR-based Re-routing* (VR) and *Preemptive-based Re-routing* (PR) techniques, which allow for fast traffic switchover upon failures without requiring ACOR synchronization or resource reservation signalling. The VR re-routes flows based on available VOPRs, and the PR preempts lower priority flows to accommodate higher priority ones. Further, the Available Reservation-based Re-routing (ARR) and Reservation Readjustment-based Re-routing (RRR) schemes are introduced for re-routing remained flows after the VR's and PR's operations. The ARR re-routes traffic after the CDPs' synchronization to overall changes occurred in network resource utilization statistics, and the RRR enables for readjusting reservations parameters on paths upon need to avoid dropping traffic unnecessarily or wasting resources upon failures. Our simulation results show that SACOR effectively provides differentiated survivability under fast convergence operations, while efficiently using the network resources.

This chapter is organized as follows. Section 6.1 describes the SACOR control approach. Section 6.2 provides the performance evaluation. Finally, the section 6.3 concludes the chapter.

6.1 *Survivable ACOR Control Approach*

The main objective of this section is to describe the SACOR approach. It describes the several phases: (A) link failure is automatically detected and announced; (B) CDPs perform the Automatic Traffic Re-routing using the VR and the PR techniques; (C) CDPs cooperate for synchronization with the changes imposed in network topology and the related resource utilization statistics; (D) CDPs with extra traffic proceed with the re-routing process through the use of ARR and RRR re-routing schemes.

6.1.1 Failure or Recovery Detection and Notification

The failure and recovery events can be detected by means of periodic hello or keep alive messages between neighboring or peering nodes, or by any technique implemented by the network administrator as in legacy systems [193], [237]. For simplicity in this work, link events "Down"/"Up" are detected based on signal power. Note that a node "Down"/"Up" event implies simultaneous failures/recoveries of multiple directly connected links. Hence, a node "Down" or "Up" event is represented as a set of links "Down" or "Up" events, with functions being processed in bundle for scalability reasons. To facilitate the understanding, the description is illustrated using Figure 6.1 with a link failure occurrence in *step (a)* between the core node C2 and CDP2.

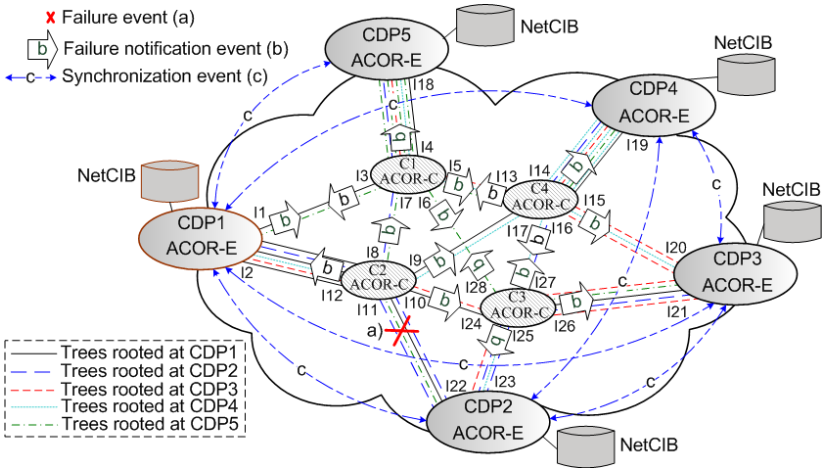


Figure 6.1. Link failure event notification message routing.

Every node which detects a link "Down" or "Up" event inside a network is responsible for announcing the occurred event to the CDPs and let the latter react and adapt to the changes required. For the sake of control stability in presence of flapping interfaces with very close "Down" and "Up" events in time, a detected "Down" event is not announced immediately as it occurs. It is announced a specified amount of time after it is detected, the *Correlation Timer* as in [237], so that unnecessary notification about a link which goes down and up within that time can be avoided. Besides, a detected "Up" event is announced after 50ms. This means that a "Down" event processing has higher priority than that of an "Up" event, since traffic must be timely re-routed in case of failures. Further details of events notification functions are provided in the following.

- After each of the nodes C2 or CDP2 (in Figure 6.1) has detected the failure of the link {C2, CDP2}, it creates a notification message and sends it out through all its outgoing interfaces as in *step (b)*. The information encapsulated in a notification message mainly includes the IDs of the affected interfaces (e.g., (I_{11}, I_{22})), the message timestamp (time at which the message was created), and a flag indicating the type of the event being announced (e.g., Down). Before sending a notification message, each of the nodes maintains a record of the event by storing the IDs of the affected interfaces, the event type and the message timestamp, and the information that is used to avoid unnecessary notification messages (as will be detailed below). A record is maintained until a configurable timer expires.

  When a notification message is traveling across a network, any visited node intercepts the message and processes it as in the following:

- First, the node checks if the message is being received for the first time by looking up in its local database whether there already exists a record of the event. In case a node receives a notification message for the first time, it records the event as explained in the previous paragraph. After that, it forwards a copy of the message on each of its interfaces except the

one on which the message was received. When there exists a record for the event (the timer has not expiered), the node simply discards the message. This ensures that each node floods a notification message about a failure or recovery event only once to avoid unnecessary flooding overhead. This procedure is repeated on every node until the messages reach the CDPs. It is also important to mention that a flooding-based solution is adopted in order to assure that a notification message is delivered to every CDP in a timely and reliable manner with less complexity, provided that there exists at least a route to the CDP. Moreover, the notification messages (control packets) are mapped to the QoS-aware control CoS maintained in ACOR for improved notification messages transport to the CDPs.

- When a CDP receives a notification message, it also forwards the message as we explained earlier. In addition, a CDP triggers the SACOR survivability processing functions so that the CDPs can quickly react and adapt to the changes occurred. In order to achieve this, survivability functions are divided into: (1) *Automatic Traffic Re-routing Functions* (ATRF) to quickly switch traffic from affected working trees to available trees, taking traffic priorities and resource availability into account; (2) *"Down" Events Synchronization Functions* (DESF) to enable CDPs to quickly synchronize with overall changes occurred in network topology and the related links resource statuses after links/nodes failures; (3) *Extra Traffic Re-routing Functions* (ETRF) to attempt to re-route the traffic flows that may not be re-routed using the ATRF functions; and (4) *"Up" Events Synchronization Functions* (UESF) to enable CDPs to quickly synchronize with changes occurred in network topology, and the related links resource statuses when previously failed links/nodes are recovered.

Notice that the timer of event record should be reasonably defined by the network administrator based on the network size to prevent unnecessary memory consumption.

6.1.2 **Automatic Traffic Re-routing Functions**

The ATRF functions enable each CDP inside a network to assure fast re-routing of traffic upon failures without requiring synchronization or resource reservation signalling. To facilitate the understanding of our description, a tree which contains failed outgoing interface(s) is called *"Failed Tree"*, and a *Failed Tree* which originates from a given CDP_A is called CDP_A's *"Own Failed Tree"*. Hence, when a CDP receives a new link failure notification message, it starts a *Failure Detection* timer so that failures that may occur within that interval can be processed in bundle for scalability reasons. When the *Failure Detection* timer expires, the CDP collects the IDs of the failed links detected, and updates the related interfaces status to "Down" in its TOPOLOGY table (see Table 6.1). Then, it obtains all the related *Failed Trees* from the *correlations pattern* in the TOPOLOGY table. Among these *Failed Trees*, it retrieves the *"Own Failed Trees"* and sets

their status to "*Failed"* in its TREES table (see Table 6.2). This way, every CDP is able to quickly distinguish its *Own Failed Trees* from the own available trees using its local database information without signalling the network. Then, the CDP attempts to re-route traffics from the *Own Failed Trees* into the own available candidate trees taking traffic priority, QoS requirements, and the available network resource into account as in the following.

First, a priority *Prio* $\in$ {high, medium, low, best effort} is assigned to each traffic flow. Moreover, traffic flows re-routing is performed aggregately for scalability reasons. This means that a set of traffic flows f (each with required bandwidth $\overline{r}_i^{\,f}$ in CoS_i) of which the aggregated used bandwidth ($\sum \overline{r}_i^{\,f}$) is smaller or equal to the available resources in CoS_i on a candidate tree are re-routed simultaneously into the tree. Further, the ATRF functions are divided into two parts as being the VR scheme and the PR techniques as follow.

- ***VOPR-based Re-routing:*** by using the VR, a CDP switches affected traffic flows of each CoS_i from each *Own Failed Tree* T_y into relevant candidate trees, which show the same egress nodes, based on the traffic priorities and available VOPRs in corresponding CoS_i on each candidate tree T_x ($\sum \overline{r}_i^{\,f} \le A_{vopr}(i, T_x)$). In particular, traffic flows removal from an *Own Failed Tree* T_y is processed using ACOR flow release functions, and the amount of used bandwidth on the tree T_y is updated in PATHS table without synchronization or QoS release signalling message into the tree T_y as we explained in subsections 4.1.2. The same way, a flow switching into a candidate tree T_x is carried out as a new flow admission into the T_x based on available VOPRs using equation (4.4). Thus, the amount of used bandwidth on the tree T_x is updated in TREES table without synchronization or QoS reservation signalling message into the tree T_x as we stated in subsection 4.1.2. Hence, the VR functions assure fast traffic re-routing, and therefore appear suitable for high priority and delay-sensitive flows.
- ***Preemptive-based Re-routing:*** The PR functions enable a CDP to remove lower priority flows from certain candidate trees in attempt to accommodate as many higher priority flows as possible without requiring signalling events, which is possible since a flow release may provide more available VOPRs in candidate trees. Therefore, the PR-based re-routing functions also provide fast traffic flows switchover, as neither synchronization nor reservation control signalling is involved. Note that the preempted traffic flows may be mapped to other trees or CoSs with lower QoS requirements depending on local control policies and service contract between customer and provider. However, the policy-driven preemption is not the focus of this work.

Table 6.1. TOPOLOGY table updating upon interface "Down" event.

Interface ID	Interface capacity	Interface sharing factor	CS	EF		AF		BE		Correlated trees per interface (*CDP ID: Tree ID*)	Interface Status	Failed correlated trees per interface (*CDP ID: Tree ID*)
				Rsv	Total used	Rsv	Total used	Rsv	Total used			
I1	1000	1	1	399.6	0	299.7	0	299.7	0	(1: 0)	Up	
I2	1000	~~3~~→2	1	399.6	0	299.7	0	299.7	0	(1: 1); (1: 2); ~~(1: 3)~~	Up	(1: 3)
I4	1000	~~4~~ → 3	1	399.6	0	299.7	0	299.7	0	(1: 0); ~~(2: 1);~~ (3: 2); (4: 3)	Up	(2: 1);
I9	1000	1	1	399.6	0	299.7	0	299.7	0	(1: 1)	Up	
I10	1000	1	1	399.6	0	299.7	0	299.7	0	(1: 2)	Up	
I11	1000	~~2~~→ 0	1	399.6	0	299.7	0	299.7	0	~~(1: 3);~~ ~~(5: 2)~~	Down	(1: 3); (5: 2)
I14	1000	4	1	399.6	0	299.7	0	299.7	0	(1: 1); (2: 2); (3: 3); (5: 0)	Up	
I26	1000	3	1	399.6	0	299.7	0	299.7	0	(1: 2); (2: 3); (5: 1);	Up	

Table 6.2. TREES Table updating upon interface "Down" event.

Tree status	Ingress CDP ID	Tree index	Egress CDP ID	Multicast channel	EF		AF		BE		*Interface_ID*: ($VOPR_{EF}$; $VOPR_{AF}$; $VOPR_{BE}$)	Remote correlated CDPs
					Used	Avail	Used	Avail	Used	Avail		
OK	CDP1	0	CDP5	(CDP1, CDP5)	0	399.6	0	299.7	0	299.7	I_1: (399.6; 299.7; 299.7); I_4: (99.9; 74.92; 74.92)	~~CDP2~~; CDP3; CDP4
OK	CDP1	1	CDP4	(CDP1, CDP4)	0	399.6	0	299.7	0	299.7	I_2: (199.8; 148.8; 148.8); I_9: (399.6; 299.7; 299.7); I_{14}: (99.9; 74.92; 74.92)	CDP2; CDP3; CDP5
OK	CDP1	2	CDP3	(CDP1, CDP3)	0	399.6	0	299.7	0	299.7	I_2: (199.8; 148.8; 148.8); I_{10}: (399.6; 299.7; 299.7); I_{26}: (133.2; 99.9; 99.9)	CDP2; CDP5
Failed	CDP1	3	CDP2	(CDP1, CDP2)	0	399.6	0	299.7	0	299.7	I_2: (133.2; 99.2; 99.2); I_{11}: (199.8; 149.85; 149.85)	~~CDP5~~

The available VOPR in formula (4.4) in candidate trees is used by VR or preempted by PR algorithms, respectively, since the VOPR enables every CDP to process as many flows as possible without requiring signalling events. When a CDP completes the ATRF process, it triggers the DESF, so that each correlated CDP can adapt its local database to the overall changes required in terms of topology and related links resource statistics, to maintain accurate control information under failures. After the CDPs are synchronized with the network overall resource conditions, the traffic flows which may still be waiting for re-routing decisions are processed. This way, SACOR aims to achieve differentiated survivability with maximum resource utilization, without dropping traffic flows unnecessarily when there are sufficient unused resources inside the network.

6.1.3 "Down" Events Synchronization Functions

This subsection describes how CDPs synchronize to changes occurred in network topology and the related resources status after failure to assure fast system convergence. To ease the understanding, an outgoing interface that belongs to a CDP_A's own tree and a *Failed Tree* is called CDP_A's *"Own Affected Interface"*, regardless of the CDP from which the failed tree originates. Hence, a CDP which experiences *Own Failed Tree(s)* upon a failure, uses the IDs of the *Own Failed Trees* to obtain all the related *Own Affected Interfaces* from its TOPOLOGY table where the *correlations patterns* provide the common interfaces between a CDP's own trees and the *Failed Trees* inside the network. Then, the CDP computes its *Aggregate Used Bandwidth* for each CoS_i on each of its *Own Affected Interfaces,* as being the total amount of bandwidth granted to the currently running traffic flows on the own trees that use the interfaces, using equation (4.3) with *m* being the number of CDP_A's own trees. After that, the CDP obtains the correlated CDPs and the related *Own Failed Trees* per each of the "*Own Affected Interfaces*" using the following algorithm. Let *h* be the number of "*Own Affected Interfaces*" and *e* being an integer, then, the process is:

Algorithm 6.1: Correlated CDPs and the related own failed trees.

/*For each Own Affected Interface I_e.*/
for *e=1:h* **do**
  (a) Get correlated CDPs of I_e.
  (b) Get the related Own Failed trees using the Correlations Information of I_e.
end

Then, the CDP advertises the *Aggregate Used Bandwidth* computed together with the IDs of the related *Own Failed Trees* to the correlated CDPs obtained using the algorithm (6.1). This way, the correlated CDPs inside a network selectively cooperate and dynamically exchange their aggregate used bandwidth statistics in each CoS on each of the "*Affected Interfaces*" and the IDs of their *Failed Trees*. Whenever a CDP receives such information and happen to realize that it has not advertised its own *Aggregate Used Bandwidth* on certain affected interfaces received from an advertisement (the CDP may not have any tree using the failed link being processed), it computes the *Aggregate Used Bandwidth* and advertises the information to the correlated CDPs. This enables each CDP to update its local database as in the following. First, each CDP computes the total amount of used bandwidth in each CoS on each of the "*Affected Interfaces*", based on the collected aggregate used bandwidths, and updates the information in its TOPOLOGY table accordingly. Moreover, it updates the correlations patterns of each *"Own Affected Interface"* by removing the *Failed Trees*' IDs from the *Correlations Patterns* into the *Failed Correlations Patterns*, and thus, the related *Interface Sharing Factors* are also updated. Besides, it updates the list of correlated CDPs in its TREES table (see Table 6.2) for each own tree which happens to use an *"Own Affected Interface"*, while the VOPRs are also updated for the interfaces using equation (4.1).

To assure that each affected CDP receives advertisements of *Aggregate Used Bandwidth* and related Failed Trees's IDs from all expected remote CDPs and properly updates its local database, every CDP maintains a list of expected correlated CDPs obtained in algorithm (6.1). Hence, a CDP is enabled to send explicit requests to any remote CDP which fails to advertise its information and expects a response within a specified time. Then, a CDP that does not advertise or reply to explicit requests until there is a timeout is considered inactive. Then, the process resumes and a warning is issued to the network administrator. As we referred earlier, the timer should be adapted by network administrator according to the size of the network, taking into account the maximum edge-to-edge round trip time inside the network. Moreover, all control messages are acknowledged for reliability reasons. As a CDP updates its local database after a failure, it attempts to re-route the eventually remained flows, *the Extra Traffics*, which were not re-routed by the ATRF.

6.1.4 Extra Traffic Re-routing Functions

The traffic flows that were not re-routed with the ATRF functions may be re-routed after the

CDPs have synchronized to the overall network resource status in subsection 6.1.3. To achieve this, a CDP_A which has extra flows to re-route collects the candidate trees of the flows and invokes the ACOR resource and admission control functions defined upon VOPRs' exhaustion as in Chapter 4. Thus, the CDP_A attempts to re-route the extra flows by using ARR techniques and RRR scheme as follows:

- ***Available Reservation-based Re-routing***: As the local database is synchronized, the flows are re-routed aggregately based on traffics priorities and the available reservations in each relevant CoS_i on each candidate tree T_x ($\sum \overline{r}_i^{\,f} \le A_{RservBW}(i, T_x)$) using equation (4.6). Then, the used bandwidth on T_x in TREES table and the total used bandwidth on the related outgoing interfaces are also updated in the TOPOLOGY table accordingly. In case all the extra traffic is re-routed based on these functions, CDP_A advertises the correlated CDPs and the network operations resume without any QoS reservation signalling.
- ***Reservation Readjustment-based Re-routing***: This function re-routes the extra traffic flows by readjusting reservation parameters of CoSs on candidate trees upon need. The RRR carries out re-routing after synchronization and reservation readjustment signalling messages to increase resource utilization efficiency upon failures. Considering that the original ACOR resource control functions process flows individually, SACOR treats the reservation readjustment for flows aggregately as in algorithm (6.2) where the ECOR resource control algorithm is described in subsection 3.2. Let *k* be the number of CoSs on each interface and *i* be an integer, the process to readjust reservation parameters on an interface I_e on a tree T_x is:

Algorithm 6.2: Aggregate reservations readjustment.

Initialization: Compute total unused bandwidth Δ_T on bottleneck interface I_b of T_x.
/*For each CoS_i on the interface I_e. */
for *i=1:k* **do**
  (a) Get aggregate used bandwidth ($r = \sum \overline{r}_i^f$) of extra flows f in CoS_i such that ($r \leqslant \Delta_T$).
  (b) Compute new reservations using r as input to the ECOR functions described in subsection 3.2.
end

As new reservation parameters are computed, they are enforced on relevant candidate trees by means of ACOR reservation signalling protocol, and the extra traffic is admitted accordingly. In case a flow cannot be re-routed after resource reservation readjustment, it is dropped since it cannot be admitted without QoS violation.

6.1.5 **"Up" Events Synchronization Functions**

When a CDP receives a new *Notification* message about a recovered link, it processes the message as in the following. First, it sets a timer to process in bundle all new recoveries that may occur within the timer. When the timer expires, it resets the recovered interfaces' status to *"Up"* in its TOPOLOGY table (see Table 6.3). Then, it uses the IDs of the recovered interfaces and the *failed correlations patterns* from the TOPOLOGY table to obtain the IDs of the previously *Own Failed Trees* to which the recovered interfaces belong. Among these trees, it retrieves its own trees which can be recovered, the *Own Recoverable Trees*, and sets their status to "*OK"* in its TREES table (see Table 6.4). In this Thesis, a *"Recoverable Tree"* is a previously *Failed Tree* of which all outgoing interfaces show "Up" status in the TOPOLOGY table. Moreover, an outgoing interface shared by a CDP_A's tree and a *Recoverable Tree* is called CDP_A's *"Own Recovery Interface"*.

Then, the CDP computes its *Aggregate Used Bandwidth* in every CoS_i on each of its *Own Recovery Interfaces* using equation (4.3), and advertises the information together with the IDs of the related *Own Recoverable Trees* to the correlated CDPs. The correlated CDPs of a given *Own Recovery Interface* are obtained using the algorithm (6.1) in subsection 6.1.3. When a CDP's "*Interface on Own Recoverable Tree"* is used by the CDP's other active tree(s) (different from the recoverable trees), the reservation parameters of each CoS on the interface are also advertised to the correlated CDPs, since these reservation parameters may have been modified in the meantime through the active trees. A CDP that does not have any tree using the recovered interfaces may be triggered by receiving such information, since it would not have advertised its aggregate used related to the recovery interfaces received. In this case, it advertises immediately its aggregate used bandwidth, and the reservations parameters in each CoS on those interfaces to the correlated CDPs, if its active trees use the interface(s).

Thus, every concerned CDP computes the total amount of used bandwidth in each CoS on each of its "*Affected Interfaces*" by using the aggregate used bandwidths collected and equation (4.3). Then, the CDP updates used bandwidth and the reservations parameters in its TOPOLOGY table accordingly. Moreover, it updates the correlations patterns of each of the interfaces by removing all the recoverable trees' IDs from the *Failed Correlations Patterns* to the *Correlations Patterns*, and thus, the *Interface Sharing Factors* are also updated accordingly. A CDP updates also the list of correlated CDPs for each own affected tree (correlated trees with the own recovery interfaces), and the VOPRs parameters on the interfaces in its TREES table using equation (4.1). In case an interface on a recovered tree is not shared by any active tree (e.g., a recovered interface), a CDP defines new reservations parameters on the interface using the initial reservation function of ACOR. Then, the CDP signals the nodes on the recoverable trees with the IDs of the concerned

outgoing interfaces, so that the new reservations are enforced on the interfaces along the trees. As in subsection 6.1.3, it is also assured that each CDP receives the *Aggregate Used Bandwidth*, the reservation parameters and IDs of recoverable trees from all expected remote CDPs to properly update its local database.

Table 6.3. TOPOLOGY table updating upon interface "Up" event.

Interface ID	Interface capacity	Interface sharing factor	CS	EF		AF		BE		Correlated trees per interface (*CDP ID: Tree ID*)	Interface status	Failed correlated trees per interface (*CDP ID: Tree ID*)
				Rsv	Total used	Rsv	Total used	Rsv	Total used			
I1	1000	1	1	399.6	0	299.7	0	299.7	0	(1: 0)	Up	
I2	1000	3 ← ~~2~~	1	399.6	0	299.7	0	299.7	0	(1: 1); (1: 2); (1: 3)	Up	~~(1: 3)~~
I4	1000	4 ← ~~3~~	1	399.6	0	299.7	0	299.7	0	(1: 0); (2: 1); (3: 2); (4: 3)	Up	~~(2: 1);~~
I9	1000	1	1	399.6	0	299.7	0	299.7	0	(1: 1)	Up	
I10	1000	1	1	399.6	0	299.7	0	299.7	0	(1: 2)	Up	
I11	1000	2 ← ~~0~~	1	399.6	0	299.7	0	299.7	0	(1: 3); (5: 2)	Up	~~(1: 3);~~ ~~(5: 2)~~
I14	1000	4	1	399.6	0	299.7	0	299.7	0	(1: 1); (2: 2); (3: 3); (5: 0)	Up	
I26	1000	3	1	399.6	0	299.7	0	299.7	0	(1: 2); (2: 3); (5: 1);	Up	

Table 6.4. TREES table updating upon interface "Up" event.

Tree status	Ingress CDP ID	Tree index	Egress CDP ID	Multicast channel	EF		AF		BE		*Interface_ID*: ($VOPR_{EF}$; $VOPR_{AF}$; $VOPR_{BE}$)	Remote correlated CDPs
					Used	Avail	Used	Avail	Used	Avail		
OK	CDP1	0	CDP5	(CDP1, CDP5)	0	399.6	0	299.7	0	299.7	I_1: (399.6; 299.7; 299.7); I_4: (99.9; 74.92; 74.92)	CDP2; CDP3; CDP4
OK	CDP1	1	CDP4	(CDP1, CDP4)	0	399.6	0	299.7	0	299.7	I_2: (133.2; 99.2; 99.2); I_9: (399.6; 299.7; 299.7); I_{14}: (99.9; 74.92; 74.92)	CDP2; CDP3; CDP5
OK	CDP1	2	CDP3	(CDP1, CDP3)	0	399.6	0	299.7	0	299.7	I_2: (133.2; 99.2; 99.2); I_{10}: (399.6; 299.7; 299.7); I_{26}: (133.2; 99.9; 99.9)	CDP2; CDP5
OK	CDP1	3	CDP2	(CDP1, CDP2)	0	399.6	0	299.7	0	299.7	I_2: (133.2; 99.2; 99.2); I_{11}: (199.8; 149.85; 149.85)	CDP5

6.2 *Performance Evaluation*

We assessed SACOR through simulation using ns-2 [233]. We focus on observing network convergence time by using the control timers, tracking the re-routing statistics and taking as basis the simulation methodology in [115] through system throughput variations and packet dropping statistics. We evaluate the statistics of successfully re-routed traffic with each of the re-routing methods for flows' differentiation: VR, PR, ARR, and RRR. We have also analyzed the system overall packets delay. In terms of network resource utilization efficiency, SACOR does not drop any flow when there is sufficient unused bandwidth on the bottleneck links of candidate trees inside the networks. As we referred in section 6.1.4, it achieves this based on the resource control functions of ACOR demonstrated in Chapter 4.

6.2.1 **Simulation Scenario**

Each of our simulation scenarios is carried out by using 4 randomly generated topologies (number of ingress routers ranging from 3 to 6; core routers from 5 to 15 and egress routers from 3 to 6) with different degrees of correlations on the links. One of the simulated network topology is presented in Figure 4.14. For simplicity, 4 CoSs configurations are implemented in each network interface, as in subsection 3.5.3. Then, traffic requests belonging to various CoSs and to three different traffic types, such as Constant Bit Rate, Pareto and Exponential, are uniformly generated between 128Kbps and 256Kbps, and are mapped to the ingress-egress pairs at Poisson arrival rate.

To show more accurate results, the simulation is run for each of the topologies with different seeds of random generation and mapping of requests to CoSs and ingress-egress pairs. Then, the obtained results per scenario are the mean values for all seeds and topologies with a confidence interval of 95%.

Moreover, each simulation is subject to various rates of failures at randomly selected links, and we study the behaviors of the networks in 4 scenarios under different levels of network congestion or overall resource utilization (45.74%, 65.23%, 85.47%, and 99.59%). We obtain these scenarios by setting links capacity to 35Mbps, 25Mbps, 16Mbps and 10Mbps respectively with a fixed number of 320 service requests in each scenario. Note therefore that the 320 traffic requests were sufficient to reach full (nearly 100%) resource utilization when each link capacity is set to 10Mbps in any of the 4 topologies. Also, the mean resource utilization or congestion level in a network is obtained as a mean ratio of the total used bandwidth to the link capacity on bottleneck outgoing interfaces of all trees inside the network.

Further, a link failure is automatically detected and announced by the directly connected nodes to the link, using the mechanism described in section 6.1. When a CDP receives a first failure notification message about a link, it starts the *Failure Detection* timer, which in our simulations is set to 10ms. Within this timer, each CDP collects all failure events that may occur. When this timer expires, the CDP performs the Automatic Traffic Re-routing using the VR and the PR techniques. Afterwards, the CDPs cooperate for synchronization with the changes imposed in network topology and the related resource utilization statistics. Hence, the survivability process is over after the synchronization is completed if there are no extra traffic flows to re-route. Otherwise, each CDP which has extra traffic proceeds with the process through the use of ARR and RRR re-routing schemes. We plot the network overall convergence time in each scenario under the various rates of failures on one hand. On the other hand, we plot the convergence time and the number of flows re-routed for each type of the proposed re-routing schemes (VR, PR, ARR, and RRR). Finally, we show the system overall packet delay in each scenario.

6.2.2 **Simulation Results**

Figure 6.2 shows the time it takes for networks to convergence upon failures under four congestion situations (45.74%, 65.23%, 85.47%, and 99.59%). A convergence time in these simulations is the period of time from failure detection to the time the network can resume its normal operation. Hence, this time is about 85ms and 90ms, respectively under 45.74% and 65.23% of congestion, while it turns to around 103ms and 118ms, respectively under the congestion levels of 85.47% and 99.60%. One can also observe that the system converges faster under low failure rate (e.g., 4 failures/minute) than under high rate. However, there is an upper

bound of about 118ms of convergence time as we observe on the fully congested network scenario (99.60% of congestion) based on its steady convergence time under failure rates beyond 6.

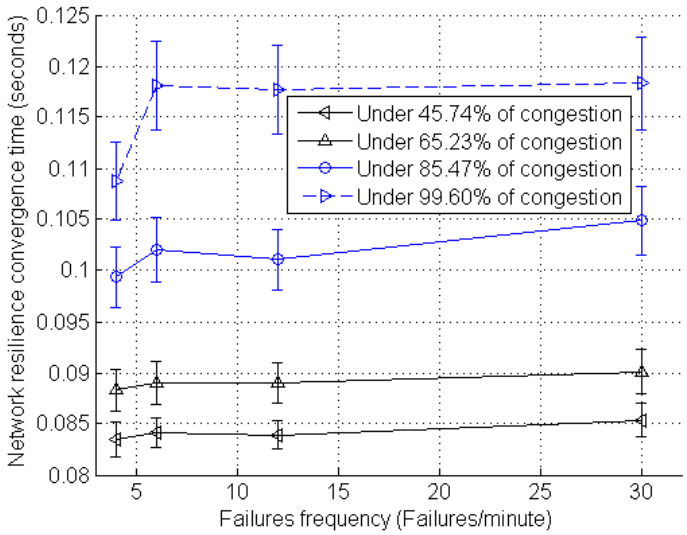


Figure 6.2. Network resilience convergence times under different levels of congestion.

In Figure 6.3, we show, for each flow re-routing scheme, the time it takes from failure detection to the time flows are re-routed. As we have seen in Figure 6.2, system convergence time increases with the increase of network congestion level. Hence, Figure 6.3 contains results for low (45.74%) and fully congested (99.60%) scenarios only for simplicity. Hence, the convergence time of about 10ms for both the VR and PR re-routing schemes show that VR and PR re-route flows automatically after the failure detection timer. Besides, flows are re-routed within about 92ms and 118ms by using ARR and RRR techniques respectively under congestion levels of 45.74% and 99.60%. In general, we observe that the ARR technique is faster than that of RRR, since the later involves resource reservation readjustment.

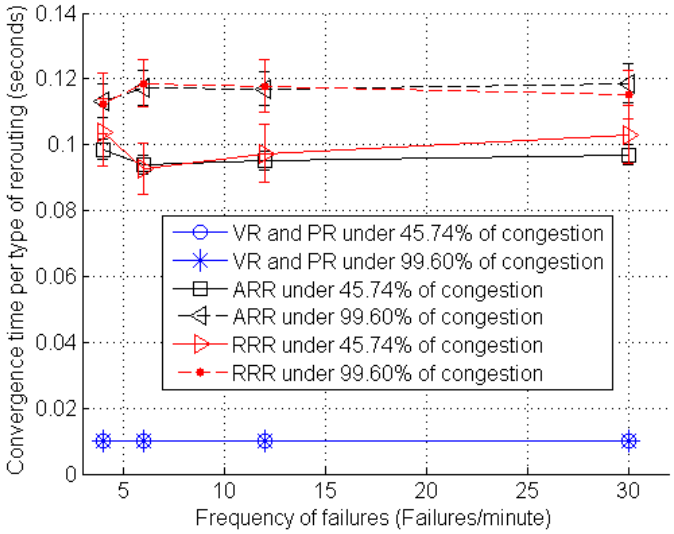


Figure 6.3. Convergence time for each type of flows re-routing.

Figure 6.4 shows the statistics of the mean number of flows re-routed per type of the proposed re-routing schemes, along with the mean number of dropped flows over the simulation time in each of the 4 scenarios. Table 6.5 summarizes the statistics to ease the observation.

Table 6.5. Summary of the statistics on Figure 6.4.

Scenarios	VR	PR	ARR	RRR	Dropped	Total
45.74 %	317	1	52	2	86	458
65.23%	278	6	60	5	88	437
85.47%	148	18	115	8	112	401
99.6%	26	27	123	14	143	333

In the first scenario (under 45.74% of congestion), 69.21% and 0.22% of affected flows were re-routed using VR and PR techniques, respectively. In other words, 69.43% of affected flows were re-routed automatically after the Failure Detection timer of 10ms using the VR and PR techniques as illustrated in Figure 6.3. In the second scenario (under 65.23% of congestion), 63.62% and 1.37% of flows were re-routed using VR and PR, respectively, while 36.91% and 4.49% were re-routed using VR and PR in the third scenario. In the last scenario under full congestion, 7.81% and 8.11% of flows were re-routed using VR and PR respectively. The percentage of flows re-routed using VR decreases, and that of the flows re-routed using PR increases with the increase of congestion level. This confirms that VOPRs' availability is inversely proportional to system congestion level. As one can notice, it was possible to re-route certain flows using VR scheme under fully congested scenario. This is due to the fact that, in shared-mesh networks, communication trees correlate dynamically by sharing resources, and affected flows may happen to release VOPRs in candidate trees upon failures. By operating based on VOPRs, and therefore without requiring synchronization or reservation signalling, the VR and PR are important to increase revenue in shared-mesh network without any dedicated protection.

Besides, we observe that 11.35% of flows are re-routed using ARR in the first scenario, and thus within about 85ms as depicted in Figure 6.3, and 0.44% of flows are re-routed using RRR within about 110ms, due to reservation readjustment signalling, and 18.78% of flows were dropped. In the second scenario, 13.73% and 1.14% of flows were re-routed using ARR and RRR respectively, and 20.14% were dropped. In the third scenario, 28.68% and 2% of flows were re-routed using ARR and RRR respectively and 27.93% were dropped. Under the fully congested scenario, 36.94% and 4.20% of flows were re-routed using ARR and RRR respectively, and 42.94% were dropped. These results show flexibility of SACOR in providing differentiated survivability control based on traffic flows' priorities, and resource availability inside a network upon failures. Recall that SACOR drops flows only when unused bandwidth is not sufficient to accommodate them without QoS violation.

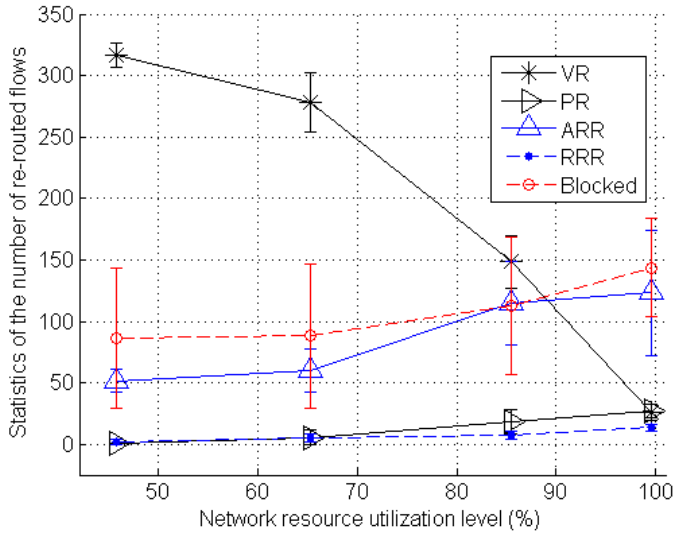


Figure 6.4. Statistics of differentiated re-routing of flows upon failures.

Figure 6.5 illustrates system overall ingress-egress packet delay per CoS in each of the four scenarios (45.74%, 65.23%, 85.47%, and 99.31%) under a scenario of 12 failures/minute. As one can expect, the packet delay is higher in more congested scenarios, but still at very low values for end-to-end.

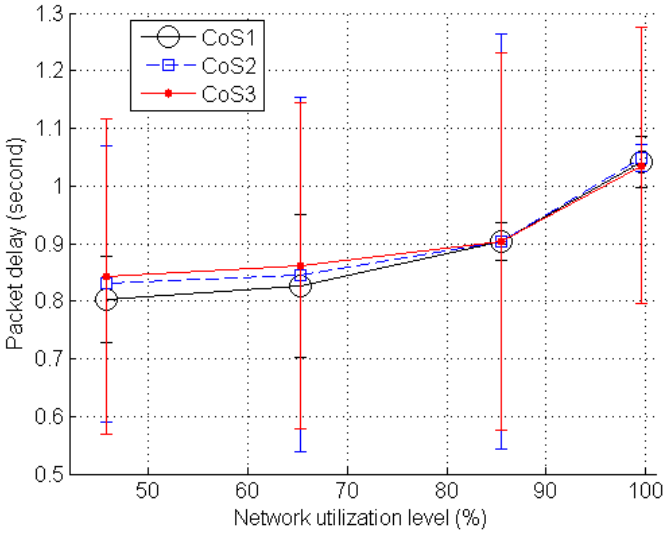


Figure 6.5. Traffic packet overall delay.

6.3 *Conclusion*

This work provided a survivability approach, the SACOR, to support stable operations in networks which implement the control approach of ACOR. SACOR pushes survivability control complexity to network CDPs, which cooperate selectively to achieve fast convergence at network borders, and interior nodes are left simpler. Moreover, it introduces automatic traffic re-routing functions using Available VOPR (VR) and Preemptive techniques (PR) to switch traffic in timely manner based on priorities without requiring signalling inside a network. Finally, SACOR provides the switching of remained flows based on ARR techniques, after synchronization of the CDPs, or based on RRR scheme to readjust reservation parameters upon need. This way, an affected flow, upon failure, is dropped only when there is no sufficient available resource on candidate paths inside the network. The obtained simulation results showed that SACOR assures fast convergence with differentiated survivability control without wasting network resources.

Chapter 7

Conclusions and Future Directions

This Thesis addressed the issues of networking control, aiming to allow for optimizing the control overall performance by taking into account key features such as Quality of Service, Scalability, Efficiency, Survivability and Cost-effectiveness. In order to achieve this, we tackled the trade-off between scalability (e.g., signalling overhead reduction) and waste of resources generally confronted in aggregate resource overprovisioning centric solutions due to dynamic characteristics of network environments. Our designs focused on a single networking control domain (e.g., area or AS), with well-defined boundaries (e.g., DiffServ or MPLS domains) where edge nodes reside (e.g., ingress/egress nodes), through which traffic may enter or exit the domain, and core nodes are placed inside the domain. This way, each network domain can deploy its own technology and inter-connect with others using any appropriate technique such as SLAs/SLSs mechanisms. In order to push control complexity to network borders and to leave core nodes simpler, we based on the generic master-client architectural elements principles, which are generally applied in the context of the NGN for policy-based control framework.

Hence, we proposed new QoS and networking control mechanisms in class-based networks with support for survivability, using aggregate resource overprovisioning concept. In our centralized design, a central entity is implemented as a CDP which takes overall control over the network, while in the decentralized solution, every edge node embeds a CDP, and all available CDPs cooperate as a means to dynamically exchange appropriate control information for synchronization to changes in network topology and related resource states. A CDP is therefore the responsible for maintaining a good knowledge of the underlying network topology and related resource conditions in real-time manner, to make policies and control decisions based on accurate information inside the network. The decisions taken by a CDP are translated into commands and

conveyed in signalling messages to the core nodes which host appropriate functions for the enforcements while being kept simpler. Further, a CDP builds multiple QoS-aware edge-to-edge multicast trees and dynamically manages aggregate bandwidth over-reservation among the CoSs configured on the trees, in a way that allows for establishing sessions without per-flow signalling for QoS or synchronization among distributed CDPs, thus achieving scalability. The use of multicast trees is a means to ensure that the packets that belonged to a flow mapped to a tree are pinned to the tree so they enjoy the QoS destined to them. We also proposed a survivability approach to support stable operations of the centralization and decentralization mechanisms introduced in this research work.

This chapter concludes the dissertation and summarizes the work done with focus on the main results achieved. It also provides open issues for further research in this area. The main contributions of this Thesis are summarized in the following.

7.1 *Summary of the Thesis*

The work carried out in this Thesis included three parts. The first part provided a general study of scalability in networking control with focus on resource overprovisioning and centralization approaches. The second part addressed the decentralization of networking control integrating overprovisioning, along with the control signalling protocol proposed for both centralized and decentralized designs. Further, the third part addressed network survivability.

In the first part, we proposed two aggregate resource over-reservation control algorithms, the COR and the ECOR. COR and ECOR provide resource computation functions which allow for dynamically defining over-reservation (surplus of reservation) parameters for CoSs configured on each outgoing interface inside a network, based on the resource conditions of the interface so that per-flow QoS signalling can be avoided. Both COR and ECOR distribute resources among CoSs on an interface using weights assigned to the CoSs in a way that prevents CoS starvation and waste of resources. However, COR relies on existing architectural control mechanism of MARA which acquires bandwidth utilization statistics using periodic and on-demand probing techniques. As a consequence, COR algorithm prevents from over-reserving too much resources per CoS by using MARA's functions, and steers focus on efficient allocation of bandwidth to avoid waste of resources when compared with MARA. In contrast, ECOR assumes a good view on network overall resource utilization statistics on real-time basis, and provides functions that allow for over-reserving to each CoS as much resources as possible, and efficiently redistribute the residual reservation in a way that also avoids wasting resources. As such, ECOR is able to allow for enhancing the overall performance.

We also proposed the ACA centralization control architecture in which a single CDP is responsible for controlling the global network. In particular, the CDP maintains in its local NetCIB database the lists of outgoing interfaces of the trees inside the network, while every session requests are sent to the CDP for AAA and admission purposes. Hence, whenever the CDP admits, terminates or readjusts a session in a CoS on a tree, it updates automatically, in its NetCIB, the resource statistics parameters of the CoS on each of the outgoing interfaces that belong to the tree. This way, the CDP maintains a good knowledge of the whole network topology, the existing trees and the related link resource usage statistics in real-time manner. The ACA implements ECOR, COR and MARA over-reservation algorithms and demonstrates that, it is possible to significantly reduce QoS control signalling, and therefore, the related processing overhead in a network without incurring QoS violations or waste of resources. We also provided a generic-purpose analytical model which allows for evaluating the impact of various control parameters (e.g., link capacity, session dynamics, etc) which generally affect the performance of over-reservation-centric approaches in terms of signalling overhead reduction and waste of resources.

In the second part of this dissertation, we proposed a generic mechanism for decentralization of network control called ACOR. The main goal of ACOR is to enable multiple CDPs distributed at network border to cooperate to exchange communication trees and related resource usage information, such that each CDP is able to maintain a good knowledge of the network topology (e.g., nodes and links) and the related links resources statistics in real-time manner. This way, ACOR provides essential support for network control sub-systems (e.g., aggregate QoS and resource over-reservations, traffic engineering, routing, and mobility), in distributed network environments. From scalability perspective, ACOR cooperation is selective, meaning that, information is exchanged between only the CDPs which are concerned and unnecessary broadcasting is avoided dynamically. Moreover, we proposed a VOPR concept which allocates a share of aggregate over-reservation of each interface to each of the trees that use the interface. Thus, the VOPR enables each CDP to process several session requests on a tree without requiring synchronization as long as the VOPR of the tree is not exhausted, such that synchronization signalling rate is also kept low. In other words, ACOR provides scalable resource and admission control functions with low QoS reservation and synchronization signalling and the related overhead without incurring QoS violation or wasting resources.

While ACOR allows for optimizing QoS reservation signalling overhead, its synchronization signalling rate increases rapidly with the increase of the number of trees that use bottleneck interfaces inside a network. Moreover, the exhaustion of VOPR upon every session request during network congestion period of time forces ACOR to place synchronization on per-request basis at congestion time, thus raised scalability concerns. Therefore, we proposed the E-ACOR. On one

hand, E-ACOR extends the ACOR VOPR concept by aggregately allocating the over-reservation of an interface to all the trees that originate at the same CDP. Hence, each CDP may process more session requests on a tree without requiring synchronization when compared with ACOR, since session demands are mostly unpredictable. On the other hand, E-ACOR enables each CDP to efficiently track network congestion information without undue control signalling overhead in a way to prevent unnecessary synchronization signalling when VOPR would exhaust under network situations. E-ACOR allows for the synchronization overhead reduction which is crucial to achieve scalability, and it is also able to keep the optimization capabilities of ACOR in terms of QoS reservation signalling overhead reduction without QoS violation or waste of resources. We also proposed the ACOR-P, an NSIS compliant signalling protocol, which defines appropriate message structures, types, fields and objects in support for all the centralization and decentralization control mechanisms designed in this Thesis. The performance evaluation through analytical and simulation results proves that E-ACOR effectively allows for optimizing network overall performance with increased resource utilization in distributed manner.

In the last part of this Thesis, we proposed the SACOR in support for stable operations and service continuity in ACOR-enabled networks in the presence of failures (e.g., links/nodes failures), or when previously failed link or new link comes up. As core nodes remain simpler, they mainly detect and announce failures or recovery events to all CDPs using a flooding-based approach. When CDPs receive notifications of failures or recoveries, they cooperate selectively to quickly adapt to the changes imposed in terms of topology, link resource conditions, and timely re-route affected traffic flows taking traffic QoS requirements and network current conditions into account. Notice that SACOR pushes survivability control load and complexity to CDPs, and core nodes are left simpler which assists to achieve fast convergence and scalability. Regarding differentiation of flows re-routing upon failures, we proposed VR and PR techniques, which allow for fast traffic switchover upon failures without requiring ACOR synchronization or resource reservation signalling. To this end, the VR re-routes flows based on available VOPRs, while the PR preempts lower priority flows to accommodate higher priority ones. Besides, we introduced the ARR and RRR schemes for re-routing remained flows after the VR's and PR's operations. The ARR re-routes traffic after the CDPs' synchronization to overall changes occurred in network resource utilization statistics, and the RRR enables for readjusting reservations parameters on trees upon need to avoid dropping traffic unnecessarily or wasting resources upon failures. Our simulation results showed that SACOR effectively supports fast convergence operations, while it allows for efficiently utilization of network resources to perform differentiated survivability.

7.2 *Future Work*

This subsection provides directions for further research and development of the work carried out in this Thesis.

In the scope of aggregate resource over-reservation algorithms, further studies in terms of context-aware assignment of weights to CoSs would allow for achieving improved results.

With respect to the centralization control framework, a prototype setup for a real testbed would also allow for a better study of the network behaviors with real data. It would allow for a more precise measurement of the performance metrics such as the signalling load, delay, loss, network resource utilization efficiency as well as QoS violation issues.

Regarding the decentralization control mechanisms of ACOR and E-ACOR, it is necessary to setup a real testbed to better evaluate the network performance in terms of signalling, state and processing overhead. This will also allow for assessing the effect of simultaneous VOPR exhaustion for further studies and analyses. The storage requirement of the NetCIB could also be evaluated. It is also of good interest to further study the approach without storing trees correlation information in local databases to assess the trade-off between bandwidth and processing consumption against storage resources utilization.

The network survivability control also needs a real testbed setup to appreciate the real measurement of network convergence time, delay, packet loss, and the performance of the flow re-routing differentiation mechanism in terms of QoE.

It is important to study inter-domain protocols including ACA, ACOR and E-ACOR architectures with efficient QoS provisioning in real scenario to evaluate the benefits and disadvantages of the proposed approaches when compared with existing Internet protocols. It would also be interesting to evaluate the performance of both centralization and decentralization in real scenarios.

Multicast trees' filtering is also an important aspect that can be further studied by taking broader performance metrics into account to improve performance.

The ACA, ACOR and E-ACOR together with the SACOR control concepts can also be applied in many different network scenarios such as network virtualization, mobility, QoS routing, Traffic engineering, and attractive service provisioning to assess the flexibility features of the approaches.

APPENDIX: ACOR-P Signalling Protocol

This annex provides details on the ACOR-P signalling protocol introduced in section 4.2 in Chapter 4.

Message Common Header

All ACOR-P messages contain the common header as the first 32 bits. It is used to specify the message type and the message relevant control flags as in Figure 1 while Table 1 provides detailed description on each field in the common header.

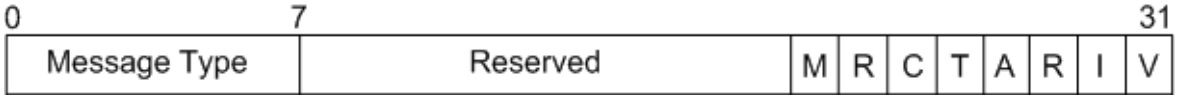


Figure 1. Common Header functional specification.

Table 1. Common header fields description.

<AttributeName>	Type	Description
Message Type	8-bit integer	- When set to 00 indicates REQUEST message. - When set to 01 indicates RESPONSE message.
VERSION (V)	1-bit (flag)	When set to TRUE=1, indicates that Internet Protocol version 6 (IPv6) is currently in use, otherwise, Internet Protocol version 4 (IPv4).
INIT (I)	1-bit (flag)	When set to TRUE=1, indicates a REQUEST message at the system initialization phase.
REFRESH (R)	1-bit (flag)	When set to TRUE=1, indicates that the amount of resources state should be refreshed in nodes as indicated in the objects.
ALTERATE (A)	1-bit (flag)	When set to TRUE=1, indicates that currently installed QSPEC states in the nodes, must be modified as indicated in the objects conveyed.
COLLECT (C)	1-bit (flag)	When set to TRUE=1, indicates that the QSPEC or MSPEC parameters must be collected on the nodes.
SYNCHRONIZE (S)	1-bit (flag)	When set to TRUE=1, indicates a synchronization message.
MULTICAST (M)	1-bit (flag)	When set to TRUE=1, indicates a request for multicast flow.

Request Identification Information

The Response Identification Information (RII) is an identifier set by a sender of a message which requires a response message. Hence, the receiver of the message must include the same RII in the corresponding response message to allow the sender to properly associate each response message to its original message. An original message may be a QoS reservation or a synchronization message. As shown in Figure 2, the RII object has a fixed length of one 32-bit integer. In ACOR, the type of RII object is set to 0x01.

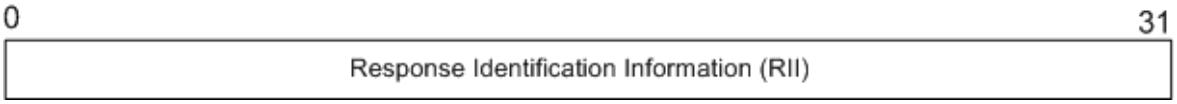


Figure 2. Response identification information functional specification.

Multicast Specification

The MSPEC object [125] is used to carry the SSM channel as a tuple (*Source_ID, Group_ID*) or a list of channels, depending on the required operation. Recall that an SSM multicast channel is characterized by two identifiers: *(1)* a *Source_ID* which indicates the ID of the media source (e.g., ingress CDP); *(2)* a *Group_ID*, which indicates the ID of the media receivers' group (e.g., egress CDP). IP address is usually used to represent the *Source_ID* or the *Group_ID* of a channel, and therefore is a 32-bit word in IPv4 or four 32-bit words in IPv6. The Figure 3 shows the format of a MSPEC object.

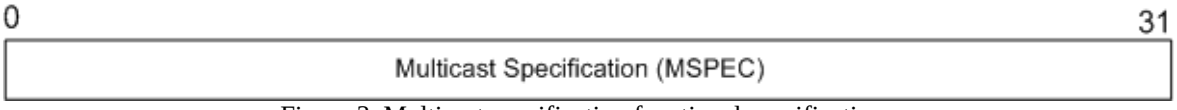


Figure 3. Multicast specification functional specification.

Record Route Object

The RRO is used to carry the list of the IDs (e.g., IP or MAC addresses) of the outgoing interfaces on a path. Hence, the size of a RRO object varies according not only to the number of the outgoing interfaces on a path, but also according to whether the IDs are IPv4, IPv6 or MAC addresses. Note that a physical (MAC) address is 48 bits for IEEE Extended Unique Identifier (EUI-48) and 64 bits for IEEE Extended Unique Identifier (EUI-64). Illustrated herein below in Figure 4, the fields of a RRO object are described in Table 2.

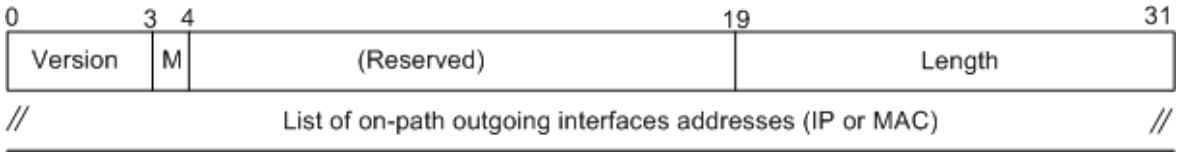


Figure 4. The RRO functional specification.

Table 2. The RRO field description.

<Attribute Name>	Type	Description
Version	4-bit (unsigned integer)	Set to 0: indicates IP address. Set to 1: indicates IEEE Extended Unique Identifier (EUI-48) MAC address. Set to 2: indicates IEEE Extended Unique Identifier (EUI-64) MAC address.
M	1-bit (flag)	Set to 1: indicates that subsequent parameter must be interpreted.
Length	12-bit (unsigned integer)	Total number of IDs in the list (variable); after adding an address, this field must increment +1.

INFO_SPEC

ACOR-P uses the INFO_SPEC object to provide feedback on a deployed operation (e.g., successful or unsuccessful notifications). As shown in Figure 5, this object contains two 16-bit integer fields, one for error code and the other for error class respectively, where the fields are described in Table 3.

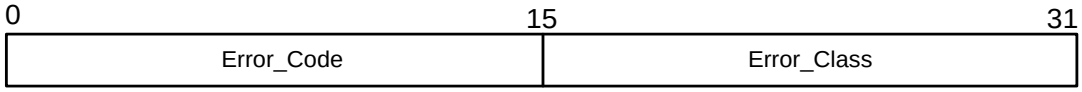


Figure 5. Info_Spec and error codes functional specification.

Table 3. INFOSPEC field description.

<AttributeName>		Description
Error_Class	Error_Code	The type is 16-bit integer for each attribute.
01	01	Possible route change occurred on downstream path.
02	01	Reservation successful.
02	02	Refreshing successful.
02	03	Modification successful.
02	04	Synchronization successful.
02	05	Notification successful.
03	01	Reservation failure: not available CoS.
03	02	Reservation failure: not available bandwidth.
03	03	Robustness failure: not enough resources.
03	04	Synchronization failure.
03	05	Modification failure.

Message ID (MSG_ID)

Every ACOR message has a unique ID. As shown in Figure 6, a message ID is a five 32-bit words field. In particular, it contains a 1-bit *Message_Binding_Type* (D) to indicate the dependency

relation of a message binding while the rest specifies a 128 bit randomly generated value that "uniquely" identifies each particular message.

When the message binding code D is set to 0: indicates unidirectional binding dependency. When the message binding code D is set to 1: indicates Bi-directional binding dependency.

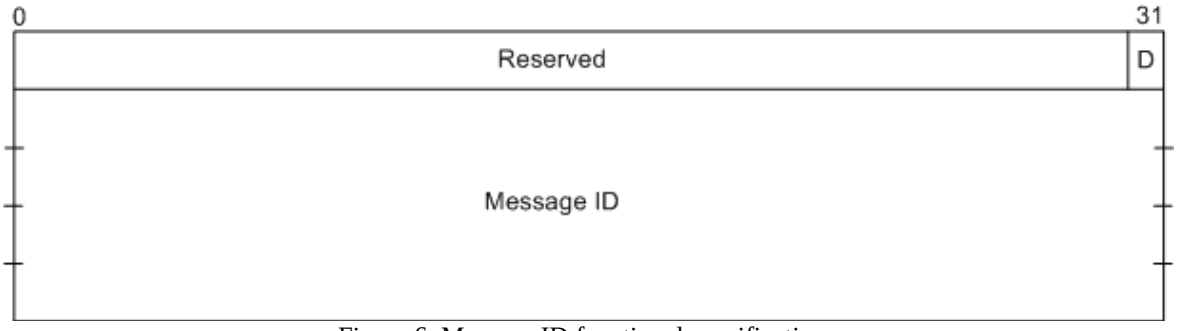


Figure 6. Message ID functional specification.

QSPEC Specification Headers

This subsection describes the format of each header used in the ACOR QSPEC containers. These headers include the QSPEC common header, the QSPEC object header and the header of each object's parameter.

QSPEC Common Header

The common header of a QSPEC object is a fixed 4-bytes long object as shown in Figure 7. It contains the version, an Initiator/Local QSPEC flag and the QSPEC Type. The version identifies the QSPEC version number assignable by IANA, while the QSPEC Type identifies the QSPEC Model deployed. Besides, an Initiator/Local QSPEC bit (I) is used to indicate whether the QSPEC is an original QSPEC initiated by the traffic source or a local QSPEC. Hence, in a domain where the control model is different from the model in the domain where the QSPEC was initiated, the initial QSPEC must be converted into the Local QSPEC to allow for a proper processing of the QSPEC across the network. Details on the format of the common header are available in Table 4.

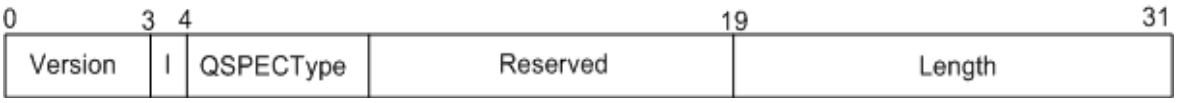


Figure 7. QSPEC Common header functional specification.

Table 4. QSPEC Common header field description.

<Attribute Name>	Type	Description
Version	4-bit (integer)	QSPEC version number assignable by IANA.
Initiator/local (I)	1-bit (flag)	Set to 0: Initiator QSPEC. Set to 1: Local QSPEC.
QSPECType	8-bit (integer)	Identification code of QoS Model in use (e.g., ACOR, E-ACOR or ACA).
Reserved	6-bit	Reserved bits.
Length	12-bit (integer)	Total length of QSPEC excluding the common header.

QSPEC Object Header

Every QSPEC object is encoded in TLV format as in Figure 8 extrated from [123] and further detailed in Table 5.

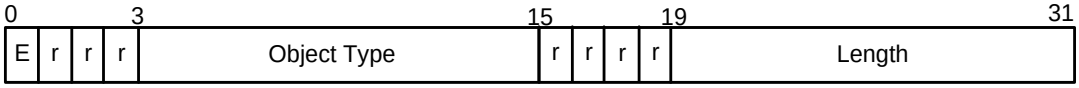


Figure 8. QSPEC object header functional specification.

Table 5. QSPEC Object header field description.

<AttributeName>	Type	Description
E	1-bit (flag)	When set to True=1, indicates error at object level.
Object Type	12-bit (integer)	Set to 0: QoS Desired (parameters cannot be overwritten). Set to 1: QoS Available (parameters may be overwritten). Set to 2: QoS Reserved (parameters cannot be overwritten). Set to 3: Minimum QoS (parameters cannot be overwritten). Set to 68: Initialization Reserve Object. Set to 69: Initialization Response to Reserve. Set to 75: Reservations Readjustment Object.
Length	12-bit (integer)	Total length of parameters excluding the object common header.

QSPEC Object Parameter Header

Each QSPEC parameter within an object is similarly encoded in TLV format using a similar parameter header as shown in Figure 9 and further detailed in Table 6.

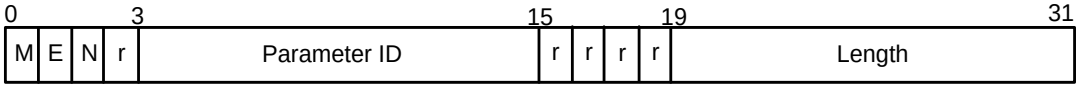


Figure 9. QSPEC object parameter header functional specification.

Table 6. QSPEC object parameter header field description.

<AttributeName>	Type	Description
M	1-bit (flag)	When set to True=1, indicates that subsequent parameter must be interpreted.
E	1-bit (flag)	When set to True=1, indicates: a) reservation failure where parameter is not met. b) error when this parameter was being interpreted.
N	1-bit (flag)	When set to True=1, indicates non-supported QSPEC parameter.
Parameter ID	12-bit (unsigned integer)	Set to 1: <TMOD-1> Set to 2: <TMOD-2> Set to 3: <Path Latency> Set to 4: <Path Jitter> Set to 5: <Path PLR> Set to 6: <Path PER> Set to 9: <Admission Priority> Set to 12: <PHB Class> Set to 14: <Y.1541 QoS Class> 260-4095—reserved IDs…. (for ACOR parameters) Set to 260: Available Bandwidth. Set to 261: Reserved Bandwidth. Set to 262: Used Bandwidth per Path. Set to 263: Aggregate Used Bandwidth. Set to 264: Aggregate VOPRs of Paths. Set to 265: Interface Capacity. Set to 266: Interface ID. Set to 267: Path ID. Set to 268: Weights of CoSs. Set to 269: Total Used Bandwidth. Set to 270: Bandwidth Threshold.
Length	12-bit (unsigned integer)	Total length of parameters excluding the object common header.

QSPEC Specification Objects

This subsection aims to provide details on the QSPEC objects used in the ACOR messages. In other words, the description introduces the format and the contents of the main QSPEC objects.

QoS Desired Object

As illustrated in Figure 10 extrated from [174], a QoS desired object contains appropriate information on the CoS and the traffic characteristics (e.g., the required bit rate, buffer size, etc). A service request message exploits this object to provide details on the QoS being requested to allow network control decision points to properly perform Admission Control and the QoS mapping.

Hence, this object may also include the QoS metrics such as the end-to-end delay, jitter and the packet loss constraints of the traffic.

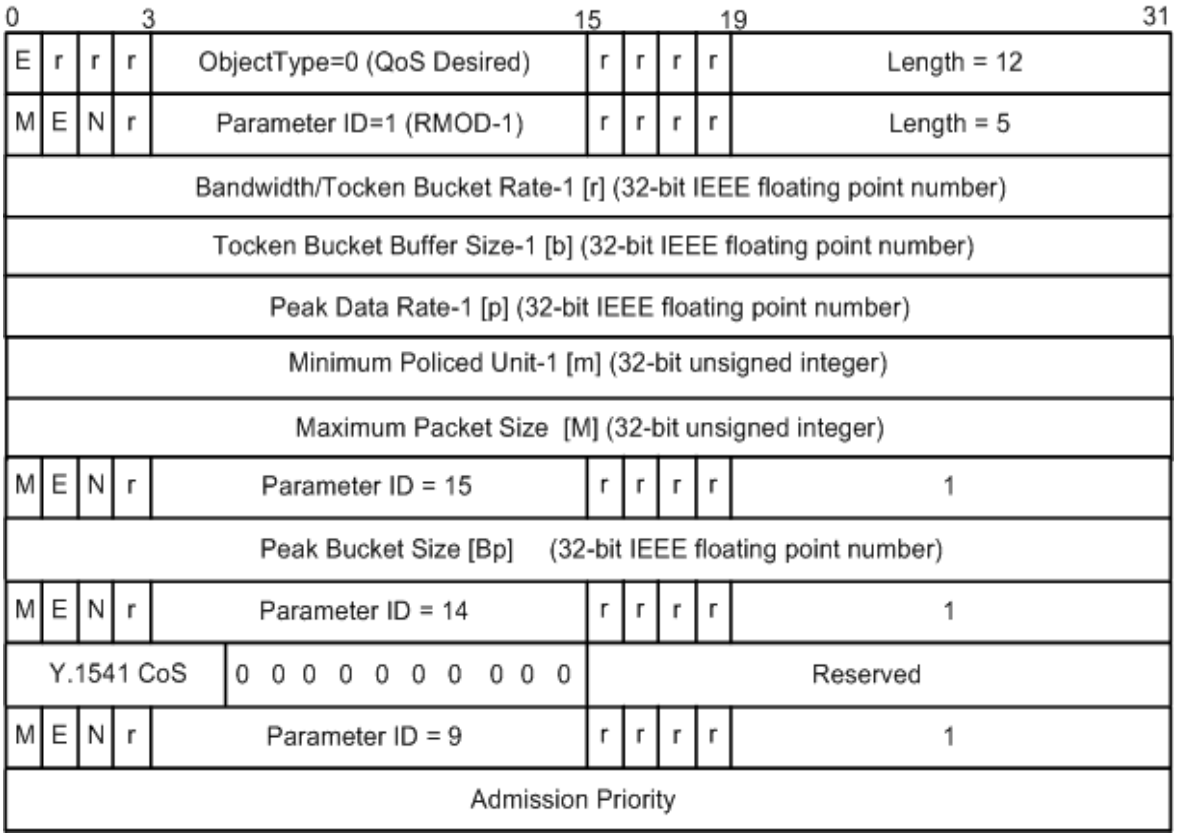


Figure 10. QoS Desired specification structure.

QoS Available Object

The QoS Available object illustrated in Figure 11 extrated from [174] is provided for the purpose of probing a communication path to collect the available resources on the bottleneck outgoing interfaces of the path. The metrics collected mainly include the available bandwidth, the end-to-end delay, jitter and packets loss on paths.

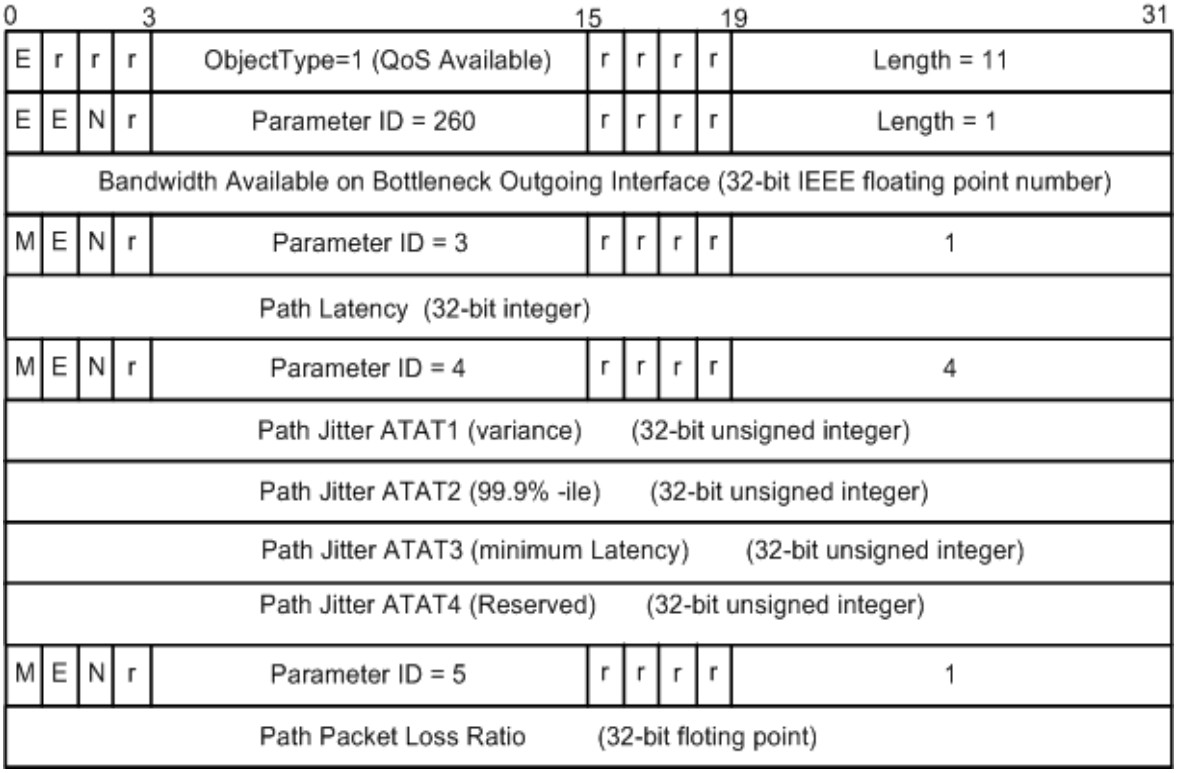


Figure 11. QoS Available Structure [Ash2010].

CXT_SPEC Common Header

As we referred earlier, the CXT_SPEC uses similar structure as that of the QSPEC. Hence, every CXT_SPEC is encoded with a common head illustrated in Figure 12 and described in Table 7.

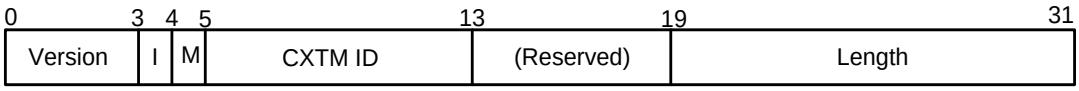


Figure 12. CXT_SPEC Common header functional specification.

Table 7. CXT_SPEC Common header field description.

<AttributeName>	**Type**	**Description**
Version	4-bit (integer)	QSPEC version number assignable by IANA
Initiator/local (I)	1-bit (flag)	Set to 0: CXT_SPEC from external CDP Set to 1: CXT_SPEC from local CDP
M	1-bit (flag)	When set to True=1, subsequent objet must be examined
CXTM ID	8-bit (integer)	Identification code of control Model: (e.g., ACOR)
Reserved	6-bit	Reserved bits
Length	12-bit (integer)	Total length of CXT_SPEC object excluding the common header.

CXT_SPEC Object Header

Figure 13 is used to illustrate the header used by each synchronization context information object and Table 8 provides details on the relevant fields' information.

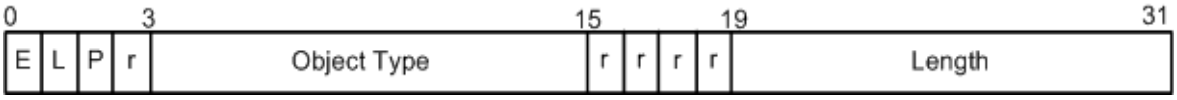


Figure 13. CXT_SPEC object header functional specification.

Table 8. CXT_SPEC Object header field description.

<AttributeName>	**Type**	**Description**
E	1-bit (flag)	When set to True=1, indicates error at object level.
L	1-bit (flag)	When set to 0, indicates Link-Down resilience event. When set to 1, indicates Link-Up resilience event.
O	1-bit (flag)	When set to 0, indicates control Option I. When set to 1, indicates control Option II.
Object Type	12-bit (integer)	Set to 70: Initial synchronization (sync.) object. Set to 71: List of paths IDs object. Set to 73: List of aggregate used bandwidth object. Set to 74: List of reservations and total used bandwidth object. Set to 75: List of reservations object. Set to 79: List of reservations, total used and thresholds object. Set to 80: List of reservations and thresholds object.
Length	12-bit (integer)	Total length of parameters excluding the object common header.

CXT_SPEC Object parameter header

Each CXT_SPEC parameter within an object as in Figure 14 is similarly encoded in TLV format using a similar parameter header as detailed in Table 9.

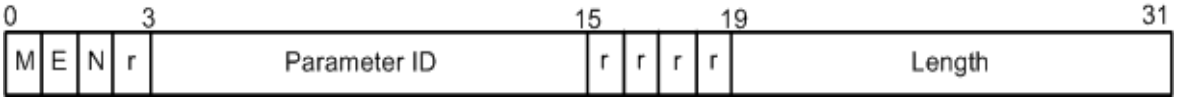


Figure 14. CXT_SPEC object parameter header functional specification.

Table 9. CXT_SPEC object parameter header field description.

<AttributeName>	Type	Description
M	1-bit (flag)	When set to True=1, indicates that subsequent parameter must be interpreted.
E	1-bit (flag)	When set to True=1, indicates error when this parameter was being interpreted.
N	1-bit (flag)	When set to True=1, indicates non-supported CXT_SPEC parameter.
Parameter ID	12-bit (unsigned integer)	Set to 260: indicates a list of bandwidth (e.g., desired, reserved, available, etc.). Set to 261: indicates a list of weights assigned to CoSs. Set to 262: indicates a list of outgoing interfaces IDs. Set to 263: indicates a list of paths IDs.
Length	12-bit (unsigned integer)	Total length of parameters excluding the object common header.

<List of bandwidths> Parameter

The <List of Bandwidths> parameter in Figure 15 is used to carry a list of bit rates information. This information can be the list of the bandwidths required to be reserved, already reserved or available bandwidth in the CoSs on a path. It is used not only for the purposes of synchronization, but also for QoS reservation and survivability control as well.

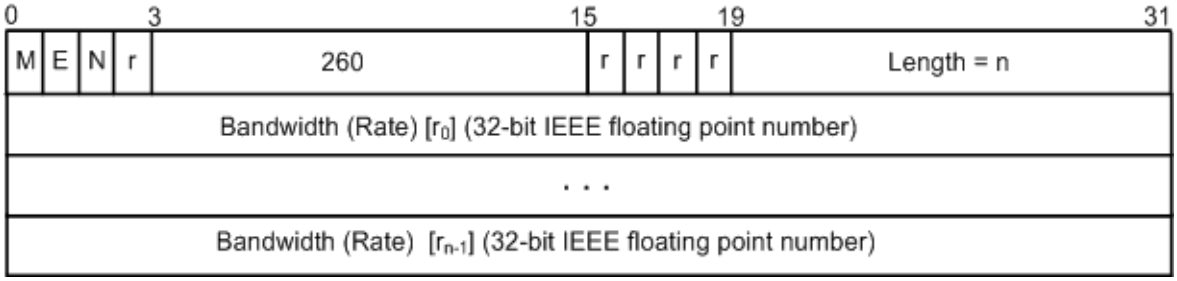


Figure 15. List of bandwidth parameter functional specification.

<List of Weights> Parameter

The <List of Weights> parameter in Figure 16 is used to carry the weights assigned to each service CoS by the administrator in the network.

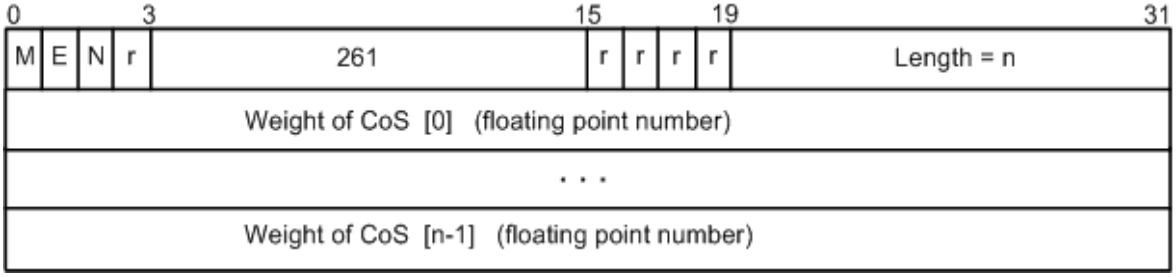


Figure 16. List of weights parameter functional specification.

<List of Outgoing Interfaces> Parameter

The <List of Outgoing Interfaces> parameter in Figure 17 is used to carry a list of outgoing interfaces such as the RRO object. It is used for synchronization as well as for survivability control.

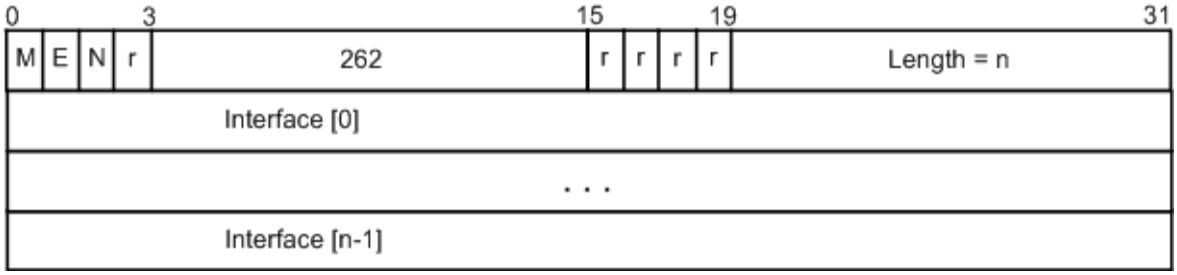


Figure 17. List of interfaces parameter functional specification.

<List of Paths> Parameter

The <List of Paths> parameter in Figure 18 is used to carry a list of communication paths IDs for the purpose of synchronization and survivability control upon need.

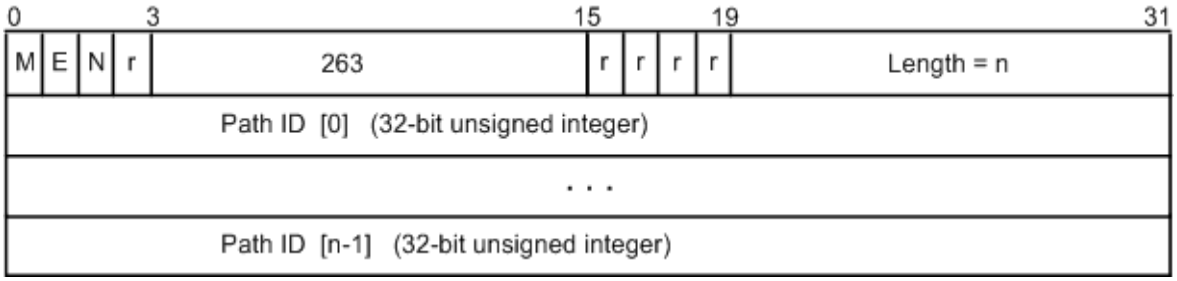


Figure 18. List of paths parameter functional specification.

CXT_SPEC Objects Type 71

The CXT_SPEC Object Type 71 as shown in Figure 19, is the object used to encapsulate paths IDs (e.g., for synchronization or survivability control). Hence, it may carry a list of selected candidate paths, lists of failed paths, list of new created paths, and so on.

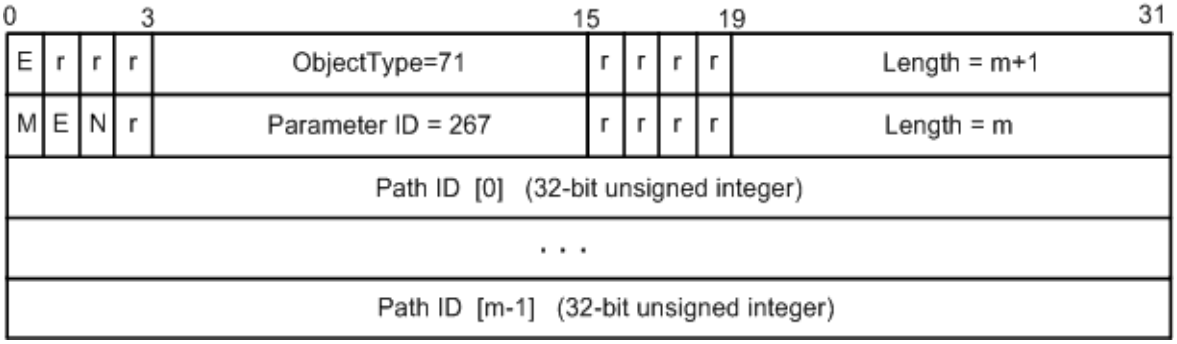


Figure 19. CXT_SPEC object type 71's functional specification.

CXT_SPEC Objects Type 73

The CXT_SPEC Object 73 as illustrated in Figure 20, is the object use to carry aggregate used bandwidth for CoSs per interface. Hence, it is used for both the synchronization and survivability control.

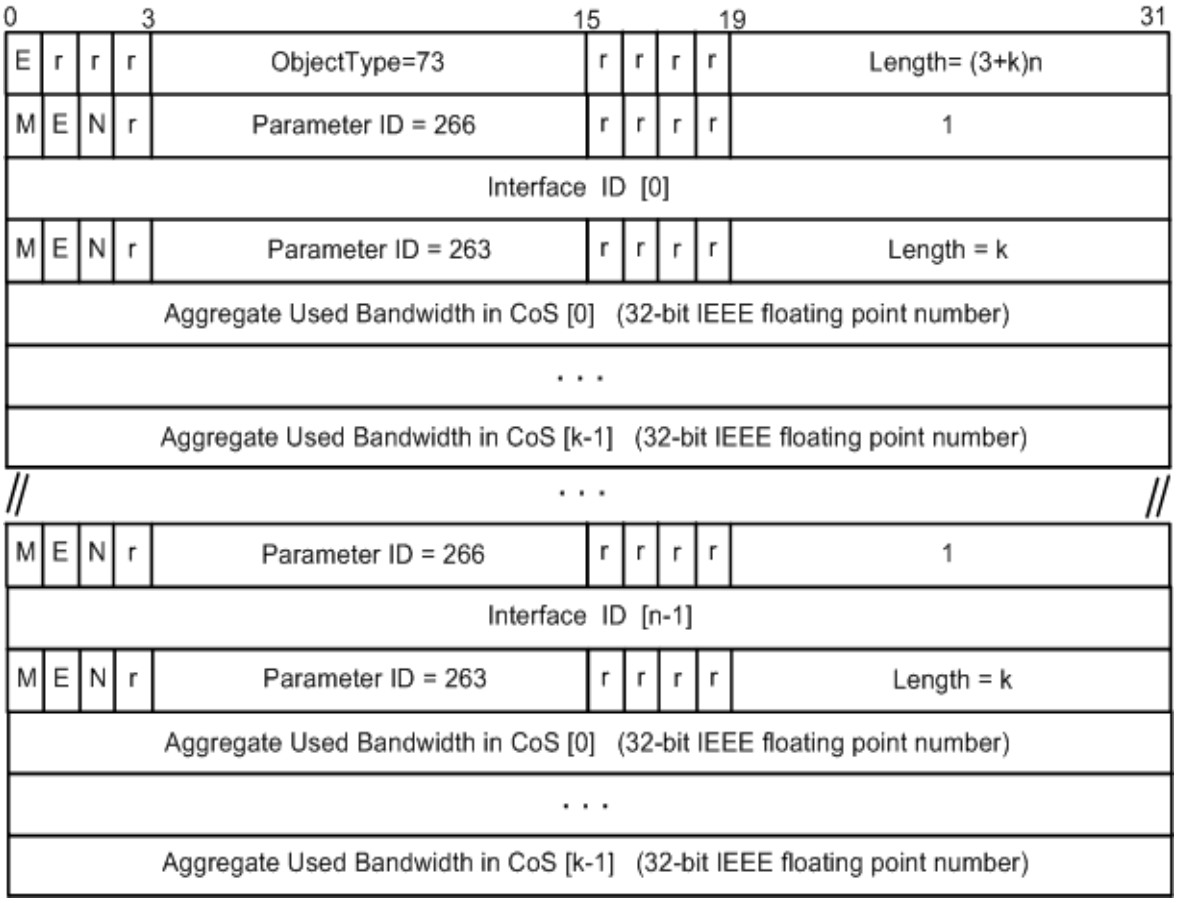


Figure 20. VOPR object (param_05) functional specification.

CXT_SPEC Objects Type 74 Specific to ACOR

The CXT_SPEC Object Type 74 illustrated in Figure 21, is the object use to encapsulate bandwidth reservation parameters and total used bandwidth parameters for CoSs on interfaces and therefore it is used for synchronization and survivability control.

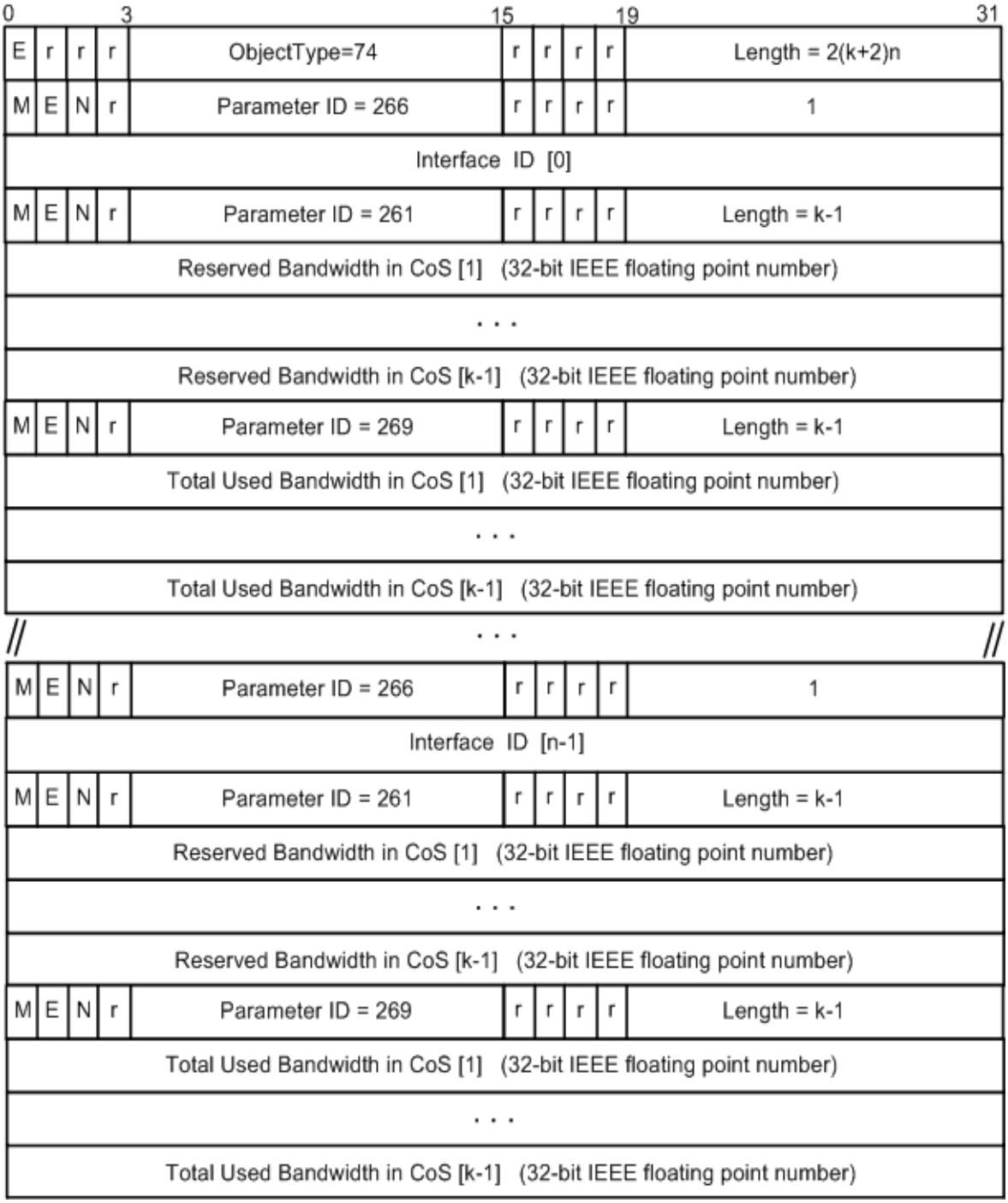


Figure 21. The CXT_SPEC Object Type 74 for ACOR.

CXT_SPEC Objects Type 74 Specific to COR or MARA

The CXT_SPEC Object Type 79 illustrated in Figure 22, is the object use to encapsulate bandwidth reservation parameters, reservation thresholds, and total used bandwidth parameters for CoSs on interfaces and for synchronization and survivability control using COR or MARA. Hence, when compared with Figure 22, one can notice that ACOR generate smaller message size than COR and MARA which is important to improve scalability.

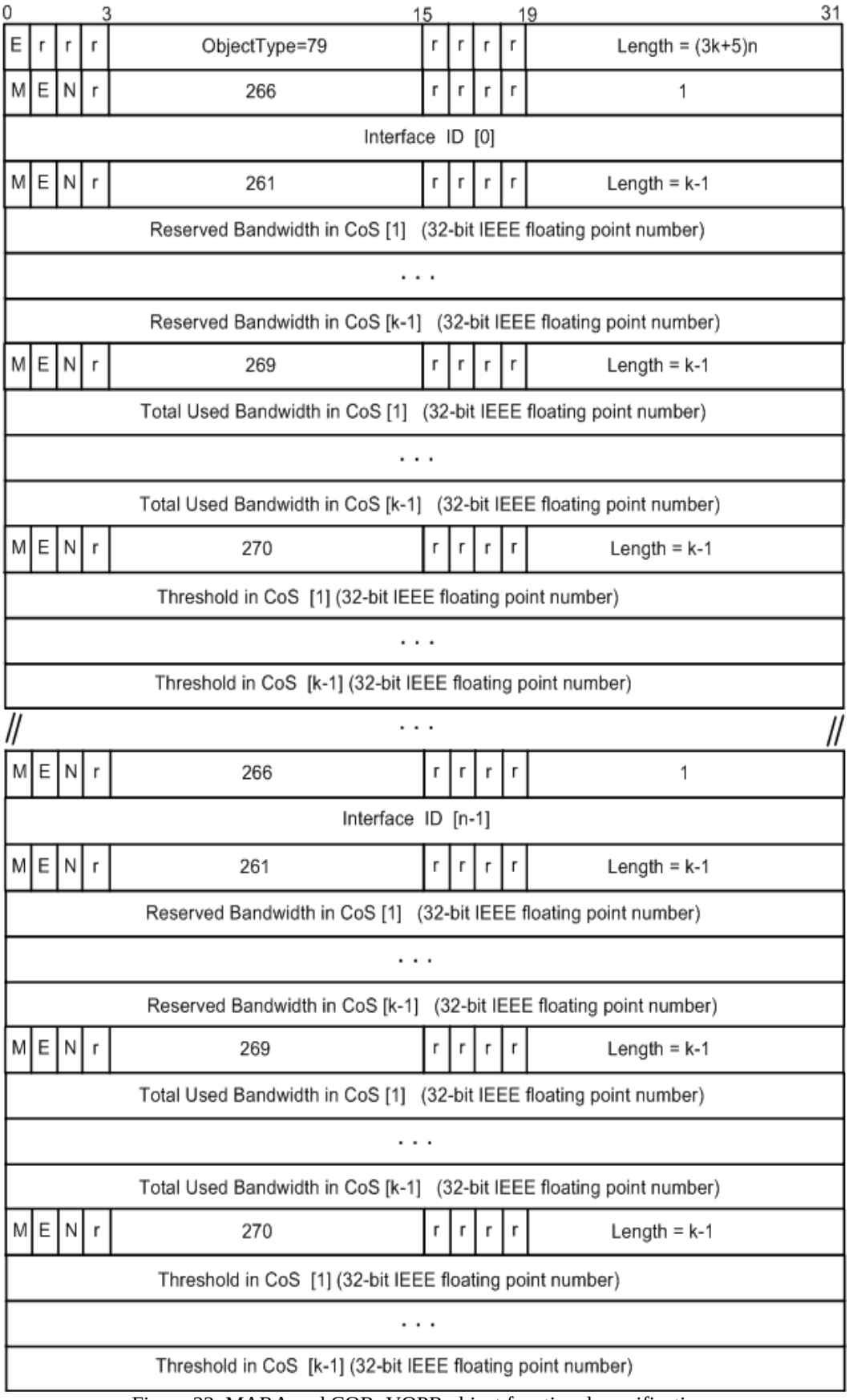


Figure 22. MARA and COR: VOPR object functional specification.

Resource Reservations Object

The resource reservation object is the QSPEC encapsulated in QoS reservation message to convey resource reservation parameters to be enforced on interfaces on a path. Figure 23 illustrates the object used in ACOR and Figure 24 illustrates the one used in COR or MARA. It becomes clear that ACOR encapsulates less information that COR or MARA and would further reduce control load.

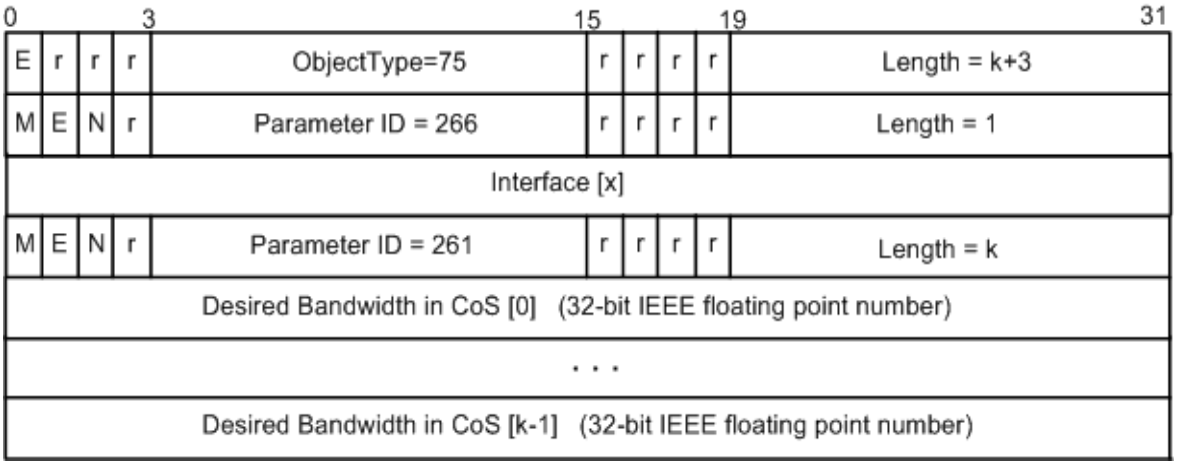


Figure 23. ACOR Resource Reservation object functional specification.

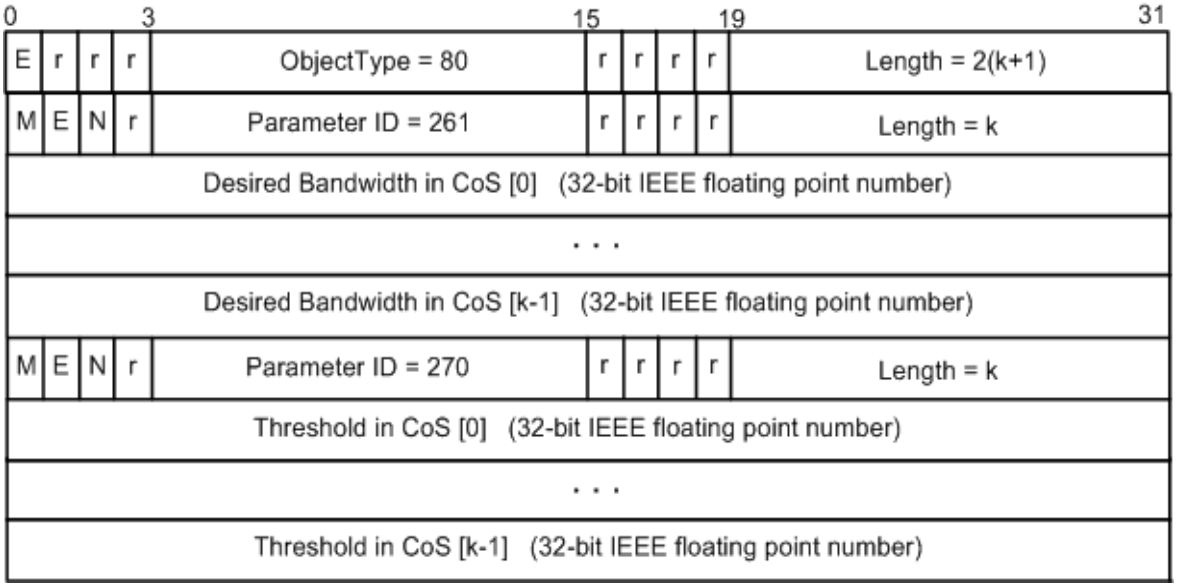


Figure 24. COR and MARA Resource Reservation object functional specification.

BIBLIOGRAPHY

[1] ITU-T Recommendation Y.2001, General overview of NGN, December 2004.

[2] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", IETF RFC 1633, June 1994.

[3] S. Blake et al, "An Architecture for Differentiated Services", IETF RFC 2475, December 1998.

[4] J. Moy, "OSPF Version 2", IETF RFC 2328, April 1998.

[5] Jun Xu, "A Survey of IP over ATM", http://www.cse.wustl.edu/~jain/cis788-97/ftp/ip_over_atm.pdf, as of August 2012.

[6] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching architecture", IETF RFC 3031, January 2001.

[7] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", IETF RFC 3945, October 2004.

[8] Cisco Systems, Inc., "Understanding MPLS-TP and Its Benefits", white paper 2009, http://www.cisco.com/en/US/technologies/tk436/tk428/white_paper_c11-562013.pdf, as of November 2012.

[9] Dieter Beller, Rolf Sperber, "MPLS-TP - The New Technology for Packet Transport Networks", DFN Forum 2009, http://www.dfn.de/fileadmin/3Beratung/DFN-Forum2/118.pdf, as of November 2012.

[10] N. Morita, H. Imanaka, O. Kamatani, T. Oba, and K. Tanida, "Overview and Status of NGN Standardization Activities at ITU-T," NTT Technical Review, Volume 5, Issue 11, 2007.

[11] Guojun Jin, Brian L. Tierney, "System Capability Effects on Algorithms for Network Bandwidth Measurement", SIGCOMM Conference, Karlsruhe, Germany, August 2003.

[12] Ningning Hu, et al.; "locating Internet Bottlenecks: Algorithms, Measurements, and Implications", SIGCOMM Conference, Portland, Oregon, USA, August-September 2004.

[13] E. Cerqueira et al. "A Unifying Architecture for Publish-Subscribe Services in the Next Generation IP Networks", IEEE Global Telecommunications Conference (IEEE GLOBECOM), San Francisco, CA, USA, November-December 2006.

[14] ITU-T Study Group 12, "Performance monitoring points for IPTV", ITUT Recommendation G.1081, October 2008.

[15] Susana Sargento, Rui Valadas; “Accurate estimation of capacities and cross-traffic of all links in a path using ICMP timestamps”, Telecommunication Systems Journal, Volume 33, Issue 1-3, Pages 89-115, December. 2006.

[16] Rito Lima S., Carvalho P., “Enabling self-adaptive QoE/QoS control”, IEEE 36th Conference on Local Computer Networks (LCN), Bonn, Germany, December 2011.

[17] Yuri Breitbart et al., “Efficient monitoring bandwidth and latency in IP networks”, Proceeding of IEEE INFOCOM, Alaska, USA, April 2001.

[18] L. Breslau, Edward W. Knigtly, S. Shenker, I. Stoica, H. Zhang, “Endpoint Admission Control: Architectural Issues and Performance”, In Proceedings of ACM SIGCOMM, Stockholm, Sweden, August-september 2000.

[19] Salehin, K.M.; Rojas-Cessa, R.; “Combined methodology for measurement of available bandwidth and link capacity in wired packet networks”, Communications Journal, IET, Volume 4, Issue 2, Pages 240 - 252, January 2010.

[20] J. Manner, X. Fu; “Analysis of Existing Quality-of-Service Signalling Protocols”, IETF RFC 4094, May 2005.

[21] Changho Yun, Harry Perros, “QoS Control for NGN: A Survey of Techniques”, Springer Journal of Network and Systems Management, Volume 18, Issue 4, Pages 447-461, December 2010.

[22] F. Baker, C. Iturralde, F. Le Faucheur, B. Davie; “Aggregation of RSVP for IPv4 and IPv6 Reservations”, IETF RFC 3175, September 2001.

[23] Kashihara, S.; Tsurusawa, M., “Dynamic Bandwidth Management System Using IP Flow Analysis for the QoS-Assured Network”, IEEE Global Telecommunications Conference (IEEE GLOBECOM), Miami, USA, December 2010.

[24] Ion Stoica, Hui Zhang, “Providing Guaranteed Services Without Per Flow Management”, In Proceedings of ACM SIGCOMM, Cambridge, MA, USA, September 1999.

[25] S. Lima, P. Carvalho, V. Freitas, “Admission Control in Multiservice IP Networks: Architectural Issues and Trends”, Topics in Network and Service Management, IEEE Communications Magazine, Volume 45, Issue 4, Page 114-121, April 2007.

[26] S. Deering, “Host Extensions for IP Multicasting”, IETF RFC 1112, August 1989.

[27] Ken Carlberg, Jon Crowcroft, “Building Shared Trees Using a One-to-Many Joining Mechanism”, in ACM SIGCOMM Computer Communication Review, Volume 27, Issue 1, Pages 5-11, January 1997.

[28] Michalis Faloutsos, Anindo Banerjea, Rajesh Pankaj, “QoSMIC: Quality of Service sensitive Multicast Internet protocol”, Proceedings of the ACM SIGCOMM, Vancouver, British Columbia, Canada, September 1998.

[29] Shigang Chen, Klara Nahrstedt, Yuval Shavitt, “A QoS-Aware Multicast Routing Protocol”, IEEE Journal on Selected Areas in Communications, Volume 18, Issue 12, December 2000.

[30] A. Neto, E. Cerqueira, A. Rissato, E. Monteiro, and P. Mendes, “A Resource Reservation Protocol Supporting QoS-aware Multicast Trees for Next Generation Networks”, 12th IEEE Symposium on Computers and Communications (ISCC), Aveiro, Portugal, July. 2007.

[31] Jun-Hong Cui, Li Lao, Michalis Faloutsos, Mario Gerla, “AQoSM: Scalable QoS Multicast Provisioning in Diff-Serv Networks”, www.cs.ucr.edu/~michalis/PAPERS/ aqosm _comnet_ 2005.pdf, as of November 2012.

[32] Joanna Moulierac, Alexandre Guitton, “QoS Scalable Tree Aggregation”, Networking, ser. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 3462, Pages 1405-1408, 2005.

[33] Fabio Mitrano et al, “QoS Management of Multicast and Broadcast Services in Next Generation Networks”, 16th IST Mobile and Wireless Communications Summit, Budapest, Hungary, July 2007.

[34] Vasco Pereira, Edmundo Monteiro, Paulo Mendes, “Evaluation of an Overlay for Source-Specific Multicast in Asymmetric Routing environments”, In Proceedings of the IEEE Global Communications Conference (IEEE GLOBECOM), Washington D.C., USA, November 2007.

[35] S. Bhattacharyya, Ed. “An Overview of Source-Specific Multicast (SSM)”, IETF RFC 3569, July 2003.

[36] A. Fei, J.-H. Cui, M. Gerla, and M. Faloutsos, “Aggregated Multicast: an approach to reduce multicast state”, Proceedings of Sixth Global Internet Symposium (GI2001), San Antonio, Texas, USA, November 2001.

[37] Jung Ha Hong, Gusak O., Sohraby K., Oliver N., “Performance Analysis of Packet Encapsulation and Aggregation”, Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Pages 137-146, September 2006.

[38] Daniel G. Waddington and Fangzhe Chang, “Realizing the Transition to IPv6”, IEEE Communications Magazine, Volume 40, Issue 6, Pages 138-147, June 2002.

[39] Z. Li, P. Mohapatra, C.-N. Chuah, “Virtual Multi-Homing: On the Feasibility of Combining Overlay Routing with BGP Routing”, University of California at Davis Technical Report: CSE-2005-2, 2005.

[40] J.Roberto Evaristo, Kevin C. Desouza, Kevin Hollister, “Centralization momentum: the pendulum swings back again”, Communications of the ACM, Volume 48, Issue 2, Pages 66-71.4, February 2005.

[41] Xiaomei Liu, Li Xiao, “A Survey of Multihoming Technology in Stub Networks: Current Research and Open Issues”, IEEE Network, Volume 21, Issue 3, Pages 32 - 40, May-June 2007.

[42] Jun Bi, Ping Hu, Lizhong Xie, “Site Multihoming: Practices, Mechanisms and Perspective”, Future Generation Communication and Networking (FGCN), Jeju-Island, Korea, December 2007.

[43] Kim, C., Caesar, M., and Rexford, J., “SEATTLE: A scalable Ethernet architecture for large enterprises”, ACM Transactions on Computer Systems (TOCS), Volume 29, Issue 1, Pages 1-35 pages, February 2011.

[44] Wakamiya, N.; Arakawa, S.; Murata, M., “Self-Organization Based Network Architecture for New Generation Networks”, First International Conference on Emerging Network Intelligence, Sliema, Malta, October 2009.

[45] S.H.L. Liang, “A New Fully Decentralized Scalable Peer-to-Peer GIS Architecture”, In Proceedings of Internatinal Society for Photogrammetry and Remote Sensing (ISPRS) XXIth Congress, Beijing, China, July 2008.

[46] T. Clausen, P. Jacquet; “Optimized Link State Routing Protocol (OLSR)”, IETF RFC 3626, October 2003.

[47] Ahmed T., et al., “Enthrone Core Networking Elements for End-to-End QoS Provision over Heterogeneous Settings”, 14th IST Mobile & Wireless Communications Summit, Dresden, Germany, June 2005.

[48] EUQOS IST project web site: http://www.euqos.org, as of May 2012.

[49] Jongtae Song, Soon Seok Lee, “Comparison of NGN QoS control Models distributed or centralized”, 14th Asia-Pacific Conference on Communications (APCC), Akihabara, Tokyo, Japan, October 2008.

[50] Hongyu Hu, Jun Bi, Tao Feng, Sen Wang, Pingping Lin, You Wang, "A Survey on New Architecture Design of Internet", International Conference on Computational and Information Sciences (ICCIS), Chengdu, Sichuan, China, October 2011.

[51] Castrucci, M.; Cecchi, M.; Priscoli, F.D.; Fogliati, L.; Garino, P.; Suraci, V., "Key concepts for the Future Internet architecture", Future Network & Mobile Summit (FutureNetw), Warsaw, Poland, June 2011.

[52] http://akari-project.nict.go.jp/eng/conceptdesign.htm, as of October 2012.

[53] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner., "OpenFlow: enabling innovation in campus networks", ACM SIGCOMM Computer Communication Review, Volume 38, Issue 2, Pages 69-74, April 2008.

[54] Global Environment for Network Innovations. Web site http://geni.net, as of November 2012.

[55] P. Cholda, A. Mykkeltveit, B. E. Helvik, O. J. Wittner, A. Jajszczyk, "A Survey of Resilience Differentiation Frameworks in Communication Networks", IEEE Communications Surveys & Tutorials, Volume 9, Issue 4, Pages 32-55, 4th Quarter 2007.

[56] A. Fumagalli and L. Valcarenghi, "IP restoration versus WDM protection: Is there an optimal choice?", IEEE Network Magazine, Volume 14, Issue 6, Pages 34–41, November 2000.

[57] H. Zhang and A. Durresi, "Differentiated Multi-layer Survivability in IP/WDM Networks", Proceedings, IEEE/IFIP Network Operations and Management Symposium (NOMS), Florence, Italy, April 2002.

[58] Tong Wang; Hui Dai; Kai Xiang; BingYu Zou, "Network system survivability survey: An evolution approach", 2nd International Conference on Future Computer and Communication (ICFCC), Wuhan, China, May 2010.

[59] Sterbenz, J.P.G., et al., "Modelling and analysis of network resilience", Third International Conference on Communication Systems and Networks (COMSNETS), Bangalore, India, January 2011.

[60] D. Oran, "OSI IS-IS intra-domain routing protocol", IETF RFC 1142, February 1990.

[61] S. Shenker et al, "Specification of Guaranteed Quality of Service", IETF RFC 2212, September 1997.

[62] J. Wroclawski, "Specification of the Controlled-load Network Element Service", IETF RFC 2211, September 1997.

[63] Braden, R., Zhang, L., Berson, S., Herzog, S., and S. Jamin, “Resource Reservation Protocol (RSVP) -- Version 1 Functional Specification”, IETF RFC 2205, September 1997.

[64] A. Demers, S. Keshav, S. Shenker; “Analysis and simulation of a fair queueing algorithm”, ACM SIGCOMM Computer Communication Review, Volume 19, Issue 4, Pages 1-12, September 1989.

[65] S.J. Golestani; “A self-clocked fair queueing scheme for broadband applications”, Proceedings IEEE INFOCOM, the Conference on Computer Communications, 13th Annual Joint Conference of the IEEE Computer and Communications Societies, Networking for Global Communications, Volume 2, Pages 636-646, Toronto, Ontario, Canada, June 1994.

[66] K. Nichols, S. Blake, F. Baker, D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, IETF RFC 2474, December 1998.

[67] J. Heinanen, F. Baker, W.Weiss, J. Wroclawski, “Assured forwarding PHB group”, IETF RFC 2597, June 1999.

[68] B. Davie, A. Charny, J. Bennet, K. Benson, J. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, “An expedited forwarding PHB”, IETF RFC 3246, March 2002.

[69] J. Loyall, A. Atlas, R. Schantz, C. Gill, D. Levine, C. O’Ryan, and D. Schmidt, “Flexible and Adaptive Control of Real-Time Distributed Object Computing Middleware”, Submitted to The International Journal of Time-Critical Computing Systems, Kluwer Academic Publishers, 2000.

[70] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang, “Recommendations on Queue Management and Congestion Avoidance in the Internet”, IETF RFC 2309, April 1998.

[71] Mao Pengxuan; Zhang Nan; Xiao Yang; Kiseon Kim, “The QOS of the Edge Router Based on Diffserv/MPLS”, 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom), Beijing, China, September 2009.

[72] W. Adis, “Quality of service middleware”, Transactions of Industrial Management & Data Systems, Volume 103, Issue 1, Pages 47-51, 2003.

[73] L. Berger, T. O’Malley, “RSVP Extensions for IPSEC Data Flows”, IETF RFC 2207, September, 1997.

[74] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, “RSVP-TE: Extensions to RSVP for LSP Tunnels”, IETF RFC 3209, December 2001.

[75] L. Berger, “Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions”, IETF RFC 3473, January 2003.

[76] Jamoussi, B., et al., “Constraint-based LSP Setup Using LDP”, IETF RFC 3212, January 2002.

[77] F. Le Faucheur, et al., “Multi-Protocol Label Switching (MPLS) Support of Differentiated Services”, IETF RFC 3270, May 2002.

[78] Gonzalo Camarillo, “Routing architecture in DiffServ MPLS networks”, Advanced Signalling Research Laboratory Ericsson, FIN-02420 Jorvas, Finland, http://www.netlab.tkk.fi/opetus/s38130/k00/Papers/Topic1-architecture.pdf, as of December 2012.

[79] R. Aggarwal, D. Papadimitriou, S. Yasukawa, “Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)”, IETF RFC 4875, May 2007.

[80] Juniper Networks, Inc., “MPLS Transport Profile (MPLS-TP) A Set of Enhancements to the Rich MPLS Toolkit”, White Paper 2011, http://www.juniper.net/us/en/local/pdf/whitepapers/2000406-en.pdf, as of September 2012.

[81] D. Katz, et al., “Traffic Engineering (TE) Extensions to OSPF Version 2”, IETF RFC 3630, September 2003.

[82] H. Smit, T. Li, “Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE)”, IETF RFC 3784, June 2004.

[83] L. Andersson, et al., “LDP Specification”, IETF RFC 5036, October 2007.

[84] Y. Rekhter, T. Li, S. Hares, “A Border Gateway Protocol 4 (BGP-4)”, IETF RFC 4271, January 2006.

[85] http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html, as of June 2012.

[86] Westberg, L. et al., “Resource Management in Diffserv (RMD) Framework”, IETF Internet Draft (draft-westberg-rmd-framework-04.tx), September 2003.

[87] S. Jamin, P. Danzig, S. Shenker, L. Zhang, “A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks”, In Proceedings of SIGCOMM, pages 2-13, Boston, MA, September 1995.

[88] D. Tse, M. Grosslauser, “Measurement-based Call Admission Control: Analysis and Simulation”, In Proceedings of INFOCOM, Pages 981-989, Kobe, Japan, April 1997.

[89] R. Prior, “Scalable Network Architecture Supporting Quality of Service”, Departamento de Ciencias de Computadores, Faculdade de Ciencias da Universidade do Porto, PhD Thesis 2007.

[90] Markus Fidler, Volker Sander, “A parameter based admission control for differentiated services networks”, Elsevier, Computer Networks Volume 4, Issue 15, Pages 463–479, March 2004.

[91] Z. Albanna, et al., “The Internet Multicast Address Allocation Architecture”, IETF RFC 3171, August 2001.

[92] http://www.youtube.com/, as of April 2012.

[93] www.bebo.com, as of September 2012.

[94] www.facebook.com, as of October 2012.

[95] http://www.microsoft.com/tv/products.mspx, as of November 2012.

[96] http://www.mmogchart.com, as of February 2012.

[97] B. Quinn, “IP Multicast Applications: Challenges and Solutions”, IETF RFC 3170, September 2001.

[98] Cain, B., Deering, S., Kouvelas, I. and A. Thyagarajan, “Internet Group Management Protocol, Version 3”, IETF RFC 3376, October 2002.

[99] Liming Wei, Deborah Estrin, “A Comparison of Multicast Trees and Algorithms”, Computer Science Department, University of Southern California, USA, Technical Report USCCS-93-560, September 1993.

[100] D. Waitzman, C. Partridge, S. Deering, “Distance Vector Multicast Routing Protocol”, IETF RFC 1075, November 1988.

[101] A. Adams, J. Nicholas, W. Siadak, “Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)”, IETF RFC: 3973, January 2005.

[102] J. Moy, “Multicast Extensions to OSPF”, IETF RFC 1584, March 1994.

[103] A. Ballardie, “Core Based Trees (CBT version 2) Multicast Routing”, IETF RFC 2189, September 1997.

[104] Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P., and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", IETF RFC 2362, June 1998.

[105] Pragyansmita Paul, S V Raghavan, "Survey of Multicast Routing Algorithms and Protocols", netlab.cs.iitm.ernet.in/publications_files/pub/pragyan/survey_multicasting.pdf, as of November 2012.

[106] Sylvia Ratnasamy, Andrey Ermolinskiy, Scott Shenker, "Revisiting IP Multicast", ACM SIGCOMM, Pisa, Italy, September 2006.

[107] T. Bates, Y. Rekhter, R. Chandra, D. Katz, "Multiprotocol Extensions for BGP-4", IETF RFC 2858, June 2000.

[108] B. Fenner, D. Meyer, "Multicast Source Discovery Protocol (MSDP)", IETF RFC 3618, October 2003.

[109] D. Thaler, "Border Gateway Multicast Protocol (BGMP): Protocol Specification", IETF RFC 3913, September 2004.

[110] P. Radoslavov, D. Estrin, R. Govindan, M. Handley, S. Kumar, D. Thaler, "The Multicast Address-Set Claim (MASC) Protocol", IETF RFC 2909, September 2000.

[111] Hugh W. Holbrook, David R. Cheriton, "IP Multicast Channels: Express Support for Large-scale Single-source Applications", in Proceedings of ACM SIGCOMM, Massachusetts, USA, August-September 1999.

[112] P. Calhoun, et al, "Diameter Base Protocol", IETF RFC 3588, September 2003.

[113] J. Case, et al., "A Simple Network Management Protocol (SNMP)", IETF RFC 1157, May 1990.

[114] K. Chan, et al., "COPS Usage for Policy Provisioning (COPS-PR)", IETF RFC 3084, March 2001.

[115] Rabat Anam Mahmood et al., "Simulating Challenges to Communication Networks for Evaluation of Resilience", Electrical Engineering & Computer Science and the Graduate Faculty of The University of Kansas School of Engineering, Master Thesis 2009.

[116] R. Yavatkar, et al., "A Framework for Policy-based Admission Control", IETF RFC 2753, January 2000.

[117] J. Rosenberg, et al., "SIP: Session Initiation Protocol", IETF RFC 3261, June 2002.

[118] M. Handley, et al., "SDP: Session Description Protocol", IETF RFC 4566, July 2006.

[119] R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch; “Next Steps in Signalling (NSIS): Framework”, IETF RFC 4080, June 2005.

[120] Schulzrinne H., R. Hancock, “GIST: General Internet Signalling Transport”, IETF Internet Draft (draft-ietf-nsis-nltp-20), June 2009.

[121] Manner, J., Karagiannis G., A. McDonald; “NSLP for Quality-of-Service Signalling”, IETF Internet Draft (draft-ietf-nsis-qos-nslp-16), February 2008.

[122] D. Katz, “IP Router Alert Option”, IETF RFC 2113, February 1997.

[123] Ash, G., Bader, A., Kappler, C., and D. Oran, “QoS NSLP QSPEC Template”, IETF Internet Draft (draft-ietf-nsis-qspec-21), November 2008.

[124] J.-P. Vasseur, Z. Ali, S. Sivabalan; “Definition of a Record Route Object (RRO) Node-Id Sub-Object”, IETF RFC 4561, June 2006.

[125] Neto A., E. Cerqueira, M. Curado, E. Monteiro and P. Mendes, “Scalable Resource Provisioning for Multi-User Communications in Next Generation Networks”, IEEE Global Telecommunications Conference (IEEE GLOBECOM), New Orleans, LA, USA, November-December 2008.

[126] Attila Bader, Georgios Karagiannis, Cornelia Kappler, Tom Phelan, “RMD-QOSM - The Resource Management in Diffserv QOS Model”, IETF Internet Draft (draft-ietf-nsis-rmd-06.txt), February 2006.

[127] Doll, M. and R. Bless, “Inter-domain Reservation Aggregation for QoS NSLP”, IETF Internet Draft (draft-bless-nsis-resv-aggr-01), July 2007.

[128] Cordeiro, L., Curado, M., Monteiro, E., Bernardo, V., Palma, D., Racaru, F., Diaz, M., and C. Chassot, “GIST Extension for Hybrid On-path Off-path Signalling (HyPath)”, IETF Internet Draft (draft-cordeiro-nsis-hypath-05), February 2008.

[129] http://www.itu.int/ITU-T/worksem/ngn/index.html - NGN, as of November 2012.

[130] hng.av.it.pt/, as of Novemver 2012.

[131] ITU-T Y.2018, “Mobility management and control framework and architecture within the NGN transport stratum”, September 2009.

[132] ITU-T Q.1706/Y.2801, “Mobility management requirements for NGN”, November 2006.

[133] ETSI ES 282 003 V3.4.2: Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control Sub-System (RACS): Functional Architecture, April 2010.

[134] ITU-T Rec. Y.2111: Resource and admission control functions in next generation networks, August 2011.

[135] 3GPP TS 23.228 V9.2.0: Technical specification group services and system aspects; IP multimedia sub-system (IMS); Stage 2 (Release 9), December 2009.

[136] TISPAN, "ETSI ES 282 001 V3.4.1: Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Functional Architecture", September 2009.

[137] ETSI TS 183 060 v0.7.1 Technical Specification, September 2008.

[138] Augusto Neto, S. Sargento, Evariste Logota, J. Antoniou, F.C Pinto, "Multiparty Session and Network Resource Control in the Context Casting (C-CAST) project", Second International Workshop on Future Multimedia Networking (FMN), Coimbra, Portugal, June 2009.

[139] T. Ahmed et al., "End-to-end Quality of Service Provisioning through an Integrated Management System for Multimedia Content Delivery", Comp. Commun., Volume 30, Issue 3, Pages 638–651, February 2007.

[140] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", IETF RFC 2638, July 1999.

[141] Z. Duan, Z.-L. Zhang, Y.T. Hou, L. Gao, "A Core Stateless Bandwidth Broker Architecture for Scalable Support of Guaranteed Services", In IEEE Transactions on Parallel and Distributed Systems, Volume 15, Issue 2, Pages 167-182, IEEE Press, February 2004.

[142] Xilouris, G.; Pliakas, T.; Kourtis, A., "Enthrone Experimental Infrastructure for E2E QOS Provisioning", IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, Athens, Greece, September 2007.

[143] Callejo-Rodrigitez, M.A., et al., "EuQoS: End-To-End QoS over Heterogeneous Networks", First ITU-T Kaleidoscope Academic Conference on Innovations in NGN: Future Network and Services (K-INGN), Geneva, Switzeland, May 2008.

[144] Alex Vallejo, Agustín Zaballos, Josep Maria Selga, and Jordi Dalmau, "Next-Generation QoS Control Architectures for Distribution Smart Grid Communication Networks", IEEE Communications Magazine, Volume 50, Issue 5, Pages 128 - 134, May 2012.

[145] Ashutosh Dutta, et al., "Self Organizing IP Multimedia Sub-system", IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA), Bangalore, India, December 2009.

[146] D. Clark, “The Design Philosophy of the DARPA Internet Protocols”, In Proceedings of ACM SIGCOMM, Stanford, CA, August 1988.

[147] Adina Magda Florea, “Self-Organizing Context Aware Agent Systems”, 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, September 2011.

[148] Chiang, F.; Braun, R., “A nature inspired multi-agent framework for autonomic service management in ubiquitous computing environments”, ICSC Congress on Computational Intelligence Methods and Applications (CIMA), Istanbul, Turkey, December 2005.

[149] C. -H. Yu, J. Werfel, R. Nagpal. “Coordinating Collective Locomotion in an Amorphous Modular Robot”, In Proceedings of International Conference on Robotics and Automation (ICRA), Anchorage, Alaska, USA, May 2010.

[150] H. Karuna et al., “Emergent Forecasting using a stigmergy approach in manufacturing coordination and control”, Engineering Self-Organising Systems, Lecture Notes in Computer Science, Volume 3464, Pages 210-226, 2005.

[151] A. Montresor, H. Meling, and O. Babaoglu, “Messor: load-balancing through a swarm of autonomous agents,” Technical Report UBLCS, Department of Computer Science University of Bologna Mura Anteo Zamboni 740127 Bologna (Italy), September 2002.

[152] N. Foukia, “IDReAM: Intrusion Detection and Response executed with Agent Mobility”, The International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Utrecht, Netherlands, July 2005.

[153] A. Kvalbein, C. Dovrolis and C. Muthu, “Multipath load-adaptive routing: putting the emphasis on robustness and simplicity,” In the Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP), Princeton, NJ, USA, October 2009.

[154] Alberto Gonzalez Prieto, et al., “Decentralized In-Network Management for the Future Internet”, IEEE International Conference on Communications Workshops, ICC Workshops, Dresden, Germany, June 2009.

[155] S. R. Madden et al., “TAG: a tiny aggregation service for ad-hoc sensor networks”, 5th Symposium on Operating Systems Design and Implementation, Boston, USA, December 2002.

[156] M. A. Sharaf et al., “Balancing energy efficiency and quality of aggregate data in sensor networks”, ACM International Journal on Very Large Data Bases, Volume 13, Issue 4, Pages 384 - 403, December 2004.

[157] M. Jelasity et al., “Gossip-based aggregation in large dynamic networks”, ACM Transactions on Computer Systems, Volume 23, Issue 3, Pages 219 – 252, August 2005.

[158] D. Kempe et al., “Gossip-Based Computation of Aggregate Information”, 44th Annual IEEE Symposium on Foundation of Computer Science (FOCS), Cambridge, USA, October 2003.

[159] Miyamura, T., et al., “Dynamic resource allocation mechanism for managed self-organization”, 13th Asia-Pacific Network Operations and Management Symposium (APNOMS), Taipei, Taiwan, September 2011.

[160] P. Psenak et al., “Multi-Topology (MT) Routing in OSPF”, IETF RFC 4915, June 2007.

[161] Y. Bernet et al., “A Framework for Integrated Services Operation over Diffserv Networks”, IETF RFC 2998, November 2000.

[162] Augusto Neto, “Multi-service Resource Allocation in the Next Generation of Networks”, University of Coimbra, Faculdade de Ciências e Tecnologia, Departamento de Engenharia Informática, PhD Thesis, 2008.

[163] L. Veloso, E. Cerqueira, P. Mendes, and E. Monteiro, “Seamless Mobility of Users with QoS and Connectivity Support”, In Proc. of the 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), New York, USA, October 2007.

[164] E. Cerqueira et al. “Multi-user Session Control in the Next Generation Wireless System”, In Proc. of the 4th ACM International Workshop on mobility Management and Wireless Access, Malaga, Spain, October 2006.

[165] P. Pan, E. Hahne, and H. Schulzrinne, “BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations”, Trans. of Communications and Networks Journal, Volume 2, Pages 157-167, June 2000.

[166] R. Sofia, R. Guerin, and P. Veiga, “SICAP, a Shared-segment Inter-domain Control Aggregation Protocol”, Workshop on High Performance Switching and Routing (HPSR), Turin, Italy, June 2003.

[167] R. Sofia, “SICAP, a Shared-segment Inter-domain Control Aggregation Protocol”, Departamento de Informática, Faculdade de Ciências da Universidade de Lisboa, Campo Grande, 1749.016 Lisboa, Portugal, PhD Thesis, March 2004.

[168] P. Pinto, A. Santos, P. Amaral, and L. Bernardo, “SIDSP: Simple Inter-domain QoS Signalling Protocol”, In Proceedings, IEEE Military Communications Conference (MILCOM), Orlando, Florida, USA, October 2007.

[169] R. Bless, “Dynamic Aggregation of Reservations for Internet Services,” Proceedings, 10th International Conference on Telecommunication Systems - Modeling and Analysis (ICTSM), Volume 1, Pages 26-38, October 2002.

[170] Rui Prior and Susana Sargento, “Scalable Reservation-Based QoS Architecture - SRBQ”, In Encyclopedia of Internet Technologies and Applications, Idea Group, Inc. (IGI) Global, Pages 473–482, October 2007.

[171] A. Neto et al., “QoS-RRC: An Overprovisioning-centric and Load Balance-aided Solution for Future Internet QoS-oriented Routing”, Springer journal - Multimedia Tools and Applications, Volume 61, Issue 3, Pages 721-746, December 2012.

[172] J. Castillo, E. Cruz, A. Neto, S. Sargento, E. Cerqueira, “Context-driven Resource Over-provisioning Approach for Rich Networking”, 21st International Conference on Computer Communications and Networks (ICCCN), Munich, Germany, July-August 2012.

[173] Context Casting - C-CAST, 7th Framework Programme, (http://www.ist-ccast.eu, as of December 2012).

[174] G. Ash, A. et al., “Y.1541-QOSM - Model for Networks Using Y.1541 QoS Classes”, IETF Internet Draft (draft-ietf-nsis-y1541-qosm-10), February 2010.

[175] Lars Westberg, András Császár, Georgios Karagiannis, Ádám Marquetant, David Partain, Octavian Pop, Vlora Rexhepi, Róbert Szabó, Attila Takács, “Resource Management in Diffserv (RMD): A Functionality and Performance Behavior Overview”, In Proceedings of the 7th International Workshop on Protocols For High-Speed Networks (PFHSN), Lecture Notes in Computer Science Volume 2334, Pages 17-34, Springer-Verlag, April 2002.

[176] A. Bader, L. Westberg, G. Karagiannis, C. Kappler, T. Phelan, “RMD-QOSM: The NSIS Quality-of-Service Model for Resource Management in Diffserv”, IETF RFC 5977, October 2010.

[177] Westberg, L., et al., “Resource Management in Diffserv On DemAnd (RODA) PHR”, IETF Internet Draft (draft-westberg-rmd-od-phr-04.txt), September 2003.

[178] A. Nucci et al., “Increasing the Link Utilization in IP over WDM Networks Using Availability as QoS”, Photonic Network Communications, Volume 9, Issue 1, Pages 55–75, January 2005.

[179] A. Shaikh and A. Greenberg, “Experience in black-box ospf measurement”, in ACM SIGCOMM Internet Measurement Workshop (IMW), San Francisco, USA, November 2001.

[180] Sandrine Pasqualini, Andreas Iselt, Andreas irstädter, and Antoine Frot, “MPLS Protection Switching vs. OSPF Re-routing A Simulative Comparison”, In Fifth International Workshop on Quality of future Internet Services (QofIS), Barcelona, Spain, September 2004.

[181] Smita Rai, Biswanath Mukherjee, Davis Omkar Deshpande, “IP Resilience within an Autonomous System: Current Approaches, Challenges, and Future Directions”, IEEE Communications Magazine, Volume 43, Issue 10, Pages 142-149, October 2005.

[182] G. Iannaccone, C. Chuah, S. Bhattacharyya, and C. Diot, “Feasibility of IP restoration in a tier-1 backbone”, In IEEE Network, Special Issue on Protection, Restoration and Disaster Recovery, March 2004.

[183] E. Oki, N. Yamanaka, and F. Pitcho, “Multiple-Availability-Level ATM Network Architecture”, IEEE Communications Magazine, Volume 33, Issue 9, Pages 80–88, September 1995.

[184] E. Ayanoglu, “A Fast Topology Update Algorithm for Restoration under Multiple Failures in Broadband Networks”, Proceedings, IEEE International Conference on Communications (ICC), pp. 1295-1299, Geneva, Switzerland, May 1993.

[185] H. Komine, T. Chujo, T. Ogura, K. Miyazaki, and T. Soejima, “A Distributed Restoration Algorithm for Multiplelink and Node Failures of Transport Networks”, In Proceedings of IEEE Global Telecommunications Conference (IEEE GLOBECOM), San Diego, USA, December 1990.

[186] R. Kawamura, K. Sato, and I Tokizawa, “Self-Healing ATM Networks Based on Virtual Path Concept,” IEEE Journal on Selected Areas in Communications, Volume 12, Issue 1, Pages 120-127, January 1994.

[187] R. Kawamura, and I. Tokizawa, “Self-healing Virtual Path Architecture in ATM Networks”, IEEE Communications Magazine, Volume 33, Issue 9, Pages 72-79, September 1995.

[188] P. Veitch, and I. Hawker, “Administration of Restorable Virtual Path Mesh Networks”, IEEE Communications Magazine, Volume 34, Issue 12, Pages 96-101, December 1996.

[189] Woungang, I., Misra, S., Obaidat, M. S., “On the Problem of Capacity Allocation and Flow Assignment in Self-Healing ATM Mesh Networks”, Computer Communications, Volume 30, Issue 16, Pages 3169-3178, November 2007.

[190] Al-Rumaih, A., Tipper, D., Liu, Y., Norman, B. A., “Spare Capacity Planning for Survivable Mesh Networks”, Proceedings of the IFIP-TC6 / European Commission International Conference on Broadband Communications, High Performance Networking, and Performance

of Communication Networks, Paris, France, Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, Volume 1815, pp. 957-968, May 2000.

[191] T. Yahara and R. Kawamura, “Virtual Path Self-Healing Scheme Based on Multi-Reliability ATM Network Concept”, In Proceedings of IEEE Global Telecommunications Conference (IEEE GLOBECOM), Phoenix, Arizona, USA, November 1997.

[192] N. Sprecher, A. Farrel, “MPLS Transport Profile (MPLS-TP) Survivability Framework”, IETF RFC 6372, September 2011.

[193] D. Papadimitriou, E. Mannie, “Analysis of Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery Mechanisms (including Protection and Restoration)”, IETF 4428, March 2006.

[194] B. Niven-Jenkins, et al., “Requirements of an MPLS Transport Profile”, IETF RFC 5654, September 2009.

[195] Christopher Metz, “IP protection and restoration”, IEEE Internet Computing, Volume 4, Issue 2, Pages 97 – 102, April 2000.

[196] J.P. Lang, et al., “RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery”, IETF RFC 4872, May 2007.

[197] J. Lang, et al., “Generalized Multi-Protocol Label Switching (GMPLS) Recovery Functional Specification”, IETF RFC 4426, March 2006.

[198] Arjan Durresi et al., “IP over All-Optical Networks - Issues”, IEEE Global Telecommunications Conference (GLOBECOM), San Antonio, USA, November 2001.

[199] Bingli Guo et al., “Dynamic Survivable Mapping in IP Over WDM Network”, Journal of Lightwave Technology, Volume 29, Issue 9, Pages 1274 - 1284, May 2011.

[200] Peera Pacharintanakul and David Tipper, “Crosslayer Survivable Mapping in Overlay-IP-WDM Networks”, 7th International Workshop on Design of Reliable Communication Networks (DRCN), Washington, D.C., October 2009.

[201] Rabindra Ghimire and Seshadri Mohan, “Design and Analysis of Protocols for QoS and Autonomous Recovery in GMPLS Controlled IP over WDM Networks”, 12th International Conference on Transparent Optical Networks (ICTON), Munich, Germany, June-July 2010.

[202] Kayi Lee et al., “Cross-Layer Survivability in WDM-Based Networks”, IEEE/ACM Transactions on Networking, Volume 19, Issue 4, Pages 1000 – 1013, August 2011.

[203] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, “Network Flows: Theory, Algorithms, and Applications”, Upper Saddle RIver, NJ: Prentice-Hall, 1993.

[204] Wojciech Molisz and Jacek Rak, “Quality of Resilience in IP-Based Future Internet Communications”, 13th International Conference on Transparent Optical Networks (ICTON), stockholm, Sweden, June 2011.

[205] Vadrevu et al., “Integrated Design for Backup Capacity Sharing Between IP and Wavelength Services in IP-Over-WDM Networks”, IEEE/OSA Journal of Optical Communications and Networking, Volume 4, Issue 1, Pages 53 – 65, January 2012.

[206] Qian Hu, Yang Wang, Xiaojun Cao, “Location-constrained Survivable Network Virtualization”, 35th IEEE Sarnoff Symposium (SARNOFF), New Brunswick, USA, April-May 2012.

[207] Yejun Liu et al., “Optimizing Backup Optical-Network-Units Selection and Backup Fibers Deployment in Survivable Hybrid Wireless-Optical Broadband Access Networks”, Journal of Lightwave Technology, Volume 30, Issue 10, Pages 1509 - 1523, May 2012.

[208] Q. She, X. Huang, and J. Jue, “Maximum survivability under multiple failures”, IEEE Optical Fiber Communication Conference and National Fiber Optic Engineers Conference, Anaheim, California, USA, March 2006.

[209] H.-W. Lee, E. Modiano, and K. Lee, “Diverse routing in networks with probabilistic failures”, IEEE/ACM Transactions on Networking, Volume 18, Issue 6, Pages 1895–1907, December 2010.

[210] Oscar Diaz et al., “Network Survivability for Multiple Probabilistic Failures”, IEEE Communications Letters, Volume 16, Issue 8, Pages 1320 - 1323, August 2012.

[211] W. Grover, et al, “Cycle-oriented Distributed Pre-Configuration Ring-Like Speed with Mesh-Like Capacity for Self-Planning Network Restoration”, IEEE International Conference on Communications (ICC), Atlanta, USA, June 1998.

[212] Matthias Baldauf, Schahram Dustdar, Florian Rosenberg, “A survey on context-aware systems”, International Journal of Ad Hoc and Ubiquitous Computing, Volume 2, Issue 4, Pages 263-277, June 2007.

[213] IST projet 507134 Ambient Networks, http://www.ambient-networks.org, as of November 2012.

[214] Abhishek Singh, Michael Conway, “Survey of Context aware Frameworks – Analysis and Criticism”, its.unc.edu/teap/tap/core/caf_review.pdf, as of September 2012.

[215] Garlan David, Siewiorek P. Daniel, Smailagic Asim, Steenkiste Peter, “Project Aura: Toward Distraction-Free Pervasive Computing”, IEEE Pervasive Computing, Volume1, Issue 2, Pages 22- 31, April-June 2002.

[216] Larry Rudolph, “Project Oxygen: Pervasive, Human-Centric Computing – An Initial Experience”, Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE), Interlaken, Switzerland, June 2001.

[217] J. Lai, A. Levas, P. Chou, C. Pinhanez, M. Viveros, “BlueSpace: Personalizing Workspace through Awareness and Adaptability”, International Journal of Human Computer Studies, Volume 57, Issue 5, Pages 415–428 November 2002.

[218] Kindberg Tim, Barton John, Morgan Jeff, Becker Gene, Caswell Debbie, Debaty Philippe, Gopal Gita, Frid Marcos, Krishnan Venky, Morris Howard, Schettino John, Serra Bill, Spasojevic Mirjana, “People, Places, Things: Web Presence for the Real World”, Third IEEE Workshop on Mobile Computing Systems and Applications, Monterey, California, USA, December 2000.

[219] http://gaia.cs.uiuc.edu/, as of June 2012.

[220] SOCAM, Tao Gu, Hung Keng Pung, Da Qing Zhang, “A Middleware for Building Context-Aware Mobile Services”, IEEE 59th Vehicular Technology Conference (VTC-Spring), Milan, Italy, May 2004.

[221] Salber Daniel, Dey A. K., Abowd G. D., “The Context Toolkit: Aiding the Development of Context-Enabled Applications”, Proceedings of the SIGCHI conference on Human Factors in Computing Systems: the CHI is the limit, ACM Press, Pittsburgh, Pennsylvania, USA, May 1999.

[222] Chen Harry, Finin Tim, Joshi Anupam, “Semantic Web in the Context Broker Architecture”, Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications (PerCom), Orlando, FL, USA, March 2004.

[223] Korpipaa, P. et al., “Managing context information in mobile devices”, IEEE Pervasive Computing, Volume 2, Issue 3, Pages 42-51, July-September 2003.

[224] Schilit B. N., Theimer M.M., “Disseminating active map information to mobile hosts”, IEEE Network: The Magazine of Global Internetworking, Volume 8, Issue 5, Pages 22-32, September-October 1994.

[225] Maarten J. van Sinderen, Aart T. van Halteren, Maarten Wegdam, Hendrik B. Meeuwissen, Eertink E. Henk, “Supporting context-aware mobile applications: an infrastructure approach”, IEEE Communications Magazine, Volume 44, Issue 9, Pages 96 – 104, September 2006.

[226] Valérie Issarny, Ferda Tartanoglu, Jinshan Liu, Françoise Sailhan, “Software Architecture for Mobile Distributed Computing”, Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA), Oslo, Norway, June 2004.

[227] Roel Ocampo, Lawrence Cheng, Kerry Jean, Alex Galis, Alberto Gonzalez Prieto, “Towards a Context Monitoring System for Ambient”, Communications and Networking in China, Beijing, China, October 2006.

[228] Hojin Lee, Bokgyun Jeon, Soyoung Park, Taekyoung Kwon, Yanghee Choi, “An Efficient Multicasting Architecture for Context-Aware Messaging Services in the Future Internet”, 10th International Conference on Advanced Communication Technology (ICACT), Phoenix Park, Korea, February 2008.

[229] Roel Ocampo, Alex Galis, Chris Todd, Hermann De Meer, “Towards Context-Based Flow Classification”, International Conference on Autonomic and Autonomous Systems (ICAS), Silicon Valley, California, USA, July 2006.

[230] R. T. Sheu and J. L. C. Wu, “A prioritized resource reservation scheme with QoS violation assessment”, Journal of Information Science and Engineering, Volume 21, Issue 1, Pages 23-37, January 2005.

[231] Yogen K., Dalal and Robert M. metcalfe; “Reverse Path Forwarding of Broadcast Packets”, Communications of the ACM, Volume 21, Issue 12, December 1978.

[232] www.javvin.com/protocolSTP.html, as of May 2012.

[233] The Network Simulator - ns-2.31, (http://www.isi.edu/nsnam/ns/, as of May 2012).

[234] J. Babiarz, K. Chan, F. Baker, “Configuration Guidelines for DiffServ Service Classes”, IETF RFC 4594, August 2006.

[235] E. Logota, A. Neto, S. Sargento, “COR: an Efficient Class-based Resource Over-pRovisioning Mechanism for Future Networks”, IEEE Symposium on Computers and Communications (ISCC), Riccione, Italy, June 2010.

[236] E. Logota, A. Neto, S. Sargento, “A New Strategy for Efficient Decentralized Network Control”, IEEE Global Telecommunications Conference (IEEE GLOBECOM), December 2010.

[237] E. Mannie, D. Papadimitriou, “Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)”, IETF RFC 4427, March 2006.

[238] Yaohui Jin, Weiqiang Sun, Wei Guo, Weisheng Hu, “Multicast Flow Aggregation in IP over WDM Networks for Large Scale Streaming Media Delivery”, International Conference on the Optical Internet (COIN) - Australian Conference on Optical Fibre Technology (ACOFT), Melbourne, Australia, June 2007.

[239] Isaac Woungang et al., “Survivability in Existing ATM-Based Mesh Networks”, International Conference on Advanced Information Networking and Applications, Bradford, United Kingdom, May 2009.